Mathematical Programming Models and Algorithms for Oshore Wind Park Design

Fischetti, Martina

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Mathematical Programming Models and Algorithms for Offshore Wind Park Design

Martina Fischetti

Kongens Lyngby 2017

Technical University of Denmark
DTU Management Engineering
Management Science
Produktionstorvet, building 424
2800 Kongens Lyngby, Denmark
Phone +45 4525 4800
www.man.dtu.dk

Summary (English)

Designing an offshore wind park is a complex process, involving several different expertises, and multiple tasks. In this thesis we developed Mathematical Programming models and algorithms to help the wind park designers. In particular, we focused on two optimization problems arising at the design phase of offshore wind parks, namely the optimal allocation of turbines in a given site and the connection of turbines through cables. We briefly touched upon the optimization of offshore jacket foundations as well.

This thesis was motivated and supervised by Vattenfall, a leading company in wind park development and operation. Thanks to our close collaboration, the optimization problems have been described and modelled as they arise in practical applications and they have been tested on real data. Our work proved to have a huge impact in practice, being able to increase park production and reduce costs. Having a sound optimization tool to help the designers allows also for different what-if analyses and scenario evaluations. This is of key value for Vattenfall, especially when looking at new technologies on the market.

The mathematical optimization models and algorithms developed have been considered of great interest also by the Operational Research (OR) community, and resulted in six journal papers. This thesis wants to follow the two-fold nature of our project, offering interesting material both to wind energy experts and practitioners, and to OR experts. Therefore we alternate OR journal papers, with practical examples and impact evaluations.

Finally, we proposed an application of integrating Machine Learning and OR,

where we investigate if a machine, trained on a large number of optimized solutions, can estimate the value of the optimized solution for new instances. This research question is of interest for all kinds of optimization problems, and is here studied on our specific wind farm application.

Resumé (Summary in Danish)

Det er en kompleks proces at designe en offshore vind park, og det involverer mange forskellige slags ekspertiser. I denne afhandling har vi udviklet matematiske programmeringsmodeller og algoritmer til at hjælpe vindparkens udviklere. Fokus har primært været på to optimeringsproblemer, der er i designfasen af en offshore vindpark, nemlig den optimale placering af vindmøller indenfor et defineret område og vindmøllernes interne forbindelser ved brug af kabler. Vi har desuden berørt optimeringen af offshore jacket fundamenter.

Denne afhandling er skrevet med inspiration og vejledning fra Vattenfall, et førende firma inden for udvikling og drift af vindparker. Takket være vores tætte samarbejde blev optimeringsproblemerne beskrevet og modelleret, som de findes i praktiske applikationer, og de er blevet testet på data fra den virkelige verden. Vores arbejde viste sig i praksis at have en signifikant indflydelse på at kunne øge parkproduktionen og reducere omkostningerne. Det udviklede optimeringsværktøj giver også mulighed for forskellige what-if analyser og scenarieevalueringer. Dette er særligt værdifuldt for Vattenfall, når nye teknologier på markedet bliver evalueret.

De udviklede matematiske optimeringsmodeller og algoritmer har også vakt interesse i interesse af Operations Research (OR) -miljøet og har resulteret i seks journal papers. Denne afhandling følger projektets todelte struktur og indeholder materiale både til vindenergieksperter/praktikere og til OR-eksperter. Derfor veksler vi mellem OR-artikler og praktiske eksempler og konsekvensevalueringer.

Endelig præsenterer vi en kombination af machine learning og OR, hvor vi undersøger om en maskine, der er trænet på mange optimerede løsninger, kan estimere værdien af den optimerede løsning i nye tilfælde. Dette spørgsmål er af interesse for alle slags optimeringsproblemer, og er her undersøgt specifikt i vindmøllekontekst.

Preface

This thesis has been developed as a collaboration between the Technical University of Denmark and Vattenfall, a leading company in the energy sector. The collaboration started during the author's master thesis (Fischetti, 2014), and was extended during the PhD project. The project has been financially supported by Vattenfall and Innovation Fund Denmark (IFD) under the Industrial PhD Program. This thesis was carried out at the Division of Management Science, DTU Management Engineering, Technical University of Denmark during the period January 2015 - December 2017. It constitutes a partial fulfilment of the requirements for acquiring a Ph.D. in Engineering. Professor David Pisinger supervised the project, and Eng. Jesper Runge Kristoffersen and Eng. Iulian Vranceanu, from Vattenfall, acted as co-supervisors.

The thesis consists of six parts. First the introduction, which gives a thorough presentation of the Offshore Wind industry, and the covered problems in the design phase of an offshore park. The next two parts are dedicated to the optimization of the turbine positions in a given site (i.e., they address the wind farm layout optimization problem) and the optimization of the cable route that connects the turbines (i.e., the inter-array cable routing problem). The fourth part is dedicated to the usage of our tools in Vattenfall, while the fifth part shows an application of Machine Learning and Operations Research together. The final part draws some conclusions and addresses future work. The thesis contains six journal papers: three published ones, two invited papers still under review, and one at the first round of review. Some chapters are currently unpublished, but we believe they can be the basis of one or more journal papers.

Lyngby, 31-December-2017

Martina Fischetti

Acknowledgements

This thesis has been developed as a collaboration between Technical University of Denmark (DTU), Management Engineering, and Vattenfall. Thanks to this very fruitful collaboration I had a unique experience, being able to investigate new Operations Research problems arising in practice. This also put me in contact with several people, with different backgrounds and experiences. I would like to thank all these colleagues, that have been always open to explain me their expertises and open to listen mine. I really think this was the key of this successful collaboration.

In particular, I would like to thank my supervisors, both from the company and from the university, who helped me in this journey.
A special "tak" goes to:
– Prof. David Pisinger for supervising my work with interest. He has always been available, and open to celebrate our successes with cookies or Porto wine. I still remember he was there at my first public talk and at my paper award.
– Eng. Jesper Runge Kristoffersen for following this project with passion. From the very beginning to the end, he has always been very dedicated and very helpful in steering the project through the most interesting directions.
– Eng. Iulian Vranceanu who has always been helpful to answer all my questions concerning cable design.

I would also like to thank Eng. Thomas Hjørt who really saw the potential of my work and pushed for testing it and routinely use it in Vattenfall. He has always been full of ideas and enthusiasm, motivating me to investigate new applications and challenges.

A special "grazie" goes of course to my family, that always supported and motivated me, to my boyfriend and to my friends, who surrounded me with joy and energy.

Contents

Summary (English) i

Resumé (Summary in Danish) iii

Preface v

Acknowledgements vii

I Introduction 1

1 Introduction 3
- 1.1 Designing competitive offshore wind parks . . . 6
- 1.2 Problem definition . . . 10
  - 1.2.1 The wind farm layout optimization problem . . . 10
  - 1.2.2 The inter-array cable routing optimization problem . . . 11
  - 1.2.3 The jacket foundation optimization problem . . . 13
- 1.3 Contribution . . . 15
- 1.4 Thesis outline . . . 17

2 Mathematical Optimization for offshore wind farm design: an overview 21
- 2.1 Introduction . . . 22
  - 2.1.1 Wind park design phases . . . 23
  - 2.1.2 Literature overview . . . 25
  - 2.1.3 Outline of the paper . . . 27
- 2.2 A Proximity Search heuristic for wind farm layout . . . 28
  - 2.2.1 Real-world application . . . 32

2.2.2 Considering cost of foundations 34
2.3 Matheuristics for cable routing 36
2.3.1 Real-world application 41
2.3.2 Cable losses 42
2.4 Jacket Foundation Optimization 45
2.4.1 The optimization model 47
2.4.2 Preliminary results 48
2.5 Conclusions 51
2.6 Future work 51

3 Matheuristics **57**
3.1 Introduction 58
3.2 General-purpose MIP-based heuristics 59
3.2.1 Local Branching 60
3.2.2 Relaxation Induced Neighborhood Search 61
3.2.3 Polishing a feasible solution 61
3.3 Proximity search heuristics for general mixed-integer programs 62
3.3.1 The basic idea 62
3.3.2 Proximity search implementations 65

II Wind Farm Layout **71**

4 Proximity search heuristics for wind farm optimal layout **73**
4.1 Introduction 74
4.2 Which MIP model? 76
4.3 Which ad-hoc heuristic? 80
4.4 Which MIP heuristic? 82
4.5 The overall approach 83
4.6 Computational results 85
4.7 Conclusions 87

5 Wind Farm Layout Optimization For Multiple Turbine Types and Multiple Zones **93**
5.1 Introduction 93
5.2 Modelling multiple turbine types 96
5.3 Tests 99
5.3.1 Different rotor diameters 100
5.3.2 Application to noise restriction 103
5.3.3 Considering the interference from existing parks 106
5.4 Conclusions 109

III Inter-array cable routing 111

6 Optimizing wind farm cable routing considering power losses 113
6.1 Introduction . . . 114
6.2 MILP model . . . 117
6.2.1 Basic model . . . 117
6.2.2 Problem complexity . . . 121
6.2.3 Improved no-cross constraints . . . 123
6.2.4 Modeling obstacles and curvy cable connections . . . 124
6.2.5 Modeling cable losses . . . 126
6.2.6 No-cross constraint separation . . . 129
6.3 Solution method . . . 130
6.3.1 A relaxed model . . . 130
6.3.2 Matheuristics . . . 131
6.3.3 Computational tuning . . . 133
6.3.4 The overall algorithm . . . 134
6.4 Computational analysis . . . 135
6.4.1 Test instances . . . 135
6.4.2 Tests . . . 137
6.5 Discussion of a real case . . . 142
6.6 Conclusions . . . 147

7 On the impact of power losses in the design of offshore wind farm cable routing 151
7.1 Introduction . . . 152
7.2 Mathematical models . . . 155
7.2.1 Basic model . . . 155
7.2.2 Cable losses . . . 157
7.2.3 Loss pre-computation . . . 160
7.3 Matheuristics . . . 161
7.3.1 A relaxed model . . . 162
7.3.2 Matheuristics based on the relaxed model . . . 163
7.4 Comparison with an existing layout . . . 165
7.5 Impact of considering power losses on real instances . . . 168
7.5.1 Test instances . . . 168
7.5.2 Impact of considering power losses . . . 169
7.6 Analysis on the energy price K . . . 172
7.6.1 Sensitivity tests on the return of investment . . . 173
7.6.2 Considerations on price fluctuations . . . 175
7.7 Conclusions . . . 178

8 Optimal wind farm cable routing: modeling branches and offshore transformer modules 183
8.1 Introduction . . . 184
8.2 The basic MILP model (OWFCR) . . . 188
8.2.1 Mathematical formulation . . . 189
8.2.2 Example . . . 191
8.3 String structure (OWFCR-SS) . . . 192
8.3.1 Mathematical formulation . . . 192
8.3.2 Example . . . 193
8.4 Branching Penalties (OWFCR-BP) . . . 193
8.4.1 Mathematical formulation . . . 194
8.4.2 Example . . . 195
8.5 Closed-loop structure (OWFCR-CL) . . . 195
8.5.1 Mathematical formulation . . . 196
8.5.2 Example . . . 197
8.6 Using OTMs instead of substations (OWFCR-OTM) . . . 198
8.6.1 Mathematical formulation . . . 198
8.6.2 Example . . . 199
8.7 A hybrid matheuristic/exact algorithm . . . 200
8.8 What-if analysis on real-world instances . . . 202
8.8.1 Test instances . . . 202
8.8.2 What-if analysis . . . 203
8.9 Conclusions . . . 207

IV Usage in Vattenfall 211

9 Testing the wind farm layout optimizer on a real project 213
9.1 Our test site . . . 214
9.2 Tests on wake effect . . . 216
9.2.1 Example of what-if analysis: exclusion of some areas . . . 219
9.3 Optimizing including cost of foundations . . . 221
9.3.1 The effect of considering cost of foundations . . . 223
9.3.2 Considering additional restrictions . . . 227

10 Usage of the optimizers in real projects 233
10.1 Introduction . . . 233
10.2 Danish Kriegers Flak . . . 235
10.3 Conclusions . . . 241

V A bit of Machine Learning 243

11 Machine Learning meets Mathematical Optimization to predict the optimal production of offshore wind parks 245
11.1 Introduction . . . 246
11.2 Basic wind energy notions . . . 250
11.2.1 Wind turbines . . . 251
11.2.2 Site and wind farm production . . . 253
11.3 The optimization model . . . 253
11.4 Data Generation . . . 256
11.4.1 Feature definition . . . 258
11.5 Machine Learning . . . 259
11.6 Results . . . 260
11.7 Conclusions and future work . . . 262

VI Closing 269

12 Conclusions and future work 271
12.1 Conclusions . . . 271
12.2 Future work . . . 274

A Computing interference 277
A.1 Jensen's model . . . 277
A.2 Jensen's model for a generic wind direction . . . 282
A.3 Weighted interference for the MILP models . . . 285
A.4 Wake effect between different turbine types . . . 289

Part I

Introduction

Chapter 1

Introduction

A transformation from fossil fuels to renewable energy has been ongoing in the last years, driven by growing environmental and sustainability demands from customers and society. Renewable energies are, indeed, inexhaustible and environmentally friendly, so they have attracted a lot of attention and investments in the last decade. An increased use of renewable sources can, in particular, help in limiting climate change, which is a very sensitive topic right now. At the Paris climate conference (COP21) in December 2015, 195 countries adopted the first-ever universal, legally binding global climate deal. The agreement sets out a global action plan to put the world on track to avoid dangerous climate changes. The ambitious targets of the Paris Agreement include also a huge increase in the use of renewable sources of energy. In order to encourage their development, many countries, such as Denmark, offers subsidies for the construction of renewable plants and warranted prices for these kinds of energy. This increased the interest in these sources of energy and, as a consequence, the competition for constructing new plants.

The increased competition implies a growing interest in reducing the Levelized Energy Cost (LEC), also known as Levelized Cost of Electricity (LCOE), for these energy sources. The LEC measure is used to compare different energy projects. It is defined as the net present value of the unit-cost of electricity over the lifetime of a generating asset. It is computed as the ratio between the costs

of a power plant and the revenue from its production. The main cost consists of CAPital EXpenditure (CAPEX), which is the immediate cost to construct the plant, and OPerational EXpenditure (OPEX), which is the cost to operate it. The production is instead often measured as Annual Energy Production (AEP), that is the MWh production of the plant over one year.

This thesis focuses on a specific type of renewable energy: wind energy. Wind energy is produced from wind farms (or wind parks) which consist of many individual wind turbines connected to the electric power transmission network. Wind parks can be located onshore (on land) or offshore (on water). This PhD project focuses on offshore wind parks. Parks located at sea have less visual/noise impact, and therefore generally use bigger/more powerful turbines. Offshore wind is also steadier and stronger than on land. Nevertheless, construction and maintenance costs are considerably higher offshore than onshore.

Figure 1.2: An offshore wind park (Ormonde): in this PhD project we optimized where to locate the turbines, how to connect them to the offshore substations and the tube selection in jacket foundations (i.e. the foundation structure on which the turbines are standing in this picture) Source: Vattenfall, 2017

In general, the LEC for wind energy (offshore in particular) drastically reduced in the last years. Offshore wind, in particular, is now facing a very fast decrease in energy prices. The first big impact in the energy prices was given by the Borssele wind park project (won by Dong Energy in 2016). The Netherlands government declared: "The cost of building and operating the Borssele offshore wind farm is expected to be Euro 2.7 billion cheaper than previously estimated. Moreover the 700 MW wind farm will generate 22.5% more electricity than anticipated. The lower than anticipated price follows fierce competition between

Figure 1.1: Constructing and operating an offshore wind farm is a complex and expensive project. Source: Vattenfall, 2017

(a) A wind turbine nacelle for the Egmond aan Zee wind farm during construction

(b) Rotor for one of the turbines at Ormonde wind farm during construction

companies in the public tender to secure the permit and associated subsidy to build and operate the wind farm." (Government of The Netherlands, July 2016) The system to decide which company should construct a wind park is indeed based on tenders: the State decides an area to construct a park and publicly provides all the site information. Different companies can compete, by proposing a wind park design for the site, together with a cost of energy. The company that can provide energy at the lowest cost wins the bid and constructs the site.

The Borssele project put a lot of pressure on all the wind energy business. In order to compete in such a market, it is now very important for companies like Vattenfall (our industrial partner) to reduce costs and increase productivity. This can be done by using cheaper components, and/or by using sound optimization tools in the different processes surrounding a new park. In this race for the lowest LEC, we worked with Vattenfall to optimize the design of offshore wind farms, in order to minimize costs and increase park production.

In 2016, Vattenfall won the Danish offshore projects Vesterhav Syd and Nord, providing a new lowest price for wind energy. Later in 2016, the cost of offshore wind has been pushed down for the third time, again by Vattenfall. Vattenfall won, indeed, the auction for Denmark's 600MW Kriegers Flak offshore wind project with a record low bid of 49.9 € per MWh (Weston, November 2016). Just in these last months (December 2017), Vattenfall is entering a subsidy-free bid for Hollandse Kust Zuid, in the Netherlands (Weston, December 2017).

1.1 Designing competitive offshore wind parks

In this thesis we focus on several optimization challenges arising in the design of an offshore wind park. We show that optimization in this phase can greatly contribute to LEC reduction for the overall project. At design phase, different experts from the developer company have to define how to design the future wind park, what components/technologies to use, and how to implement them. In order to assist them, we developed different optimizations tools, able to compare different technologies and reduce costs/increase revenues.

Our work was born from the practical need of Vattenfall to optimize the main components of its offshore wind design process.

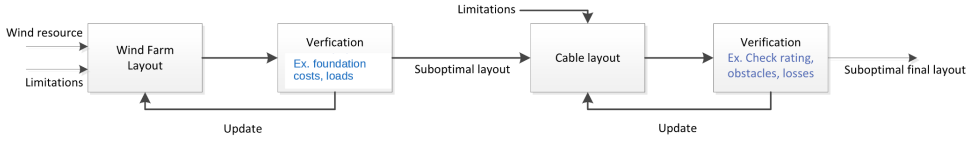
As already mentioned, the establishment of a new wind park in Europe follows this principle: first of all, the State identifies an area that is considered suitable

Figure 1.3: An offshore wind park (DanTysk) Source: Vattenfall, 2017

for a new wind park. The area is then put on tender, and different developer companies can enter it, with a detailed design for the park. The design that offers the lowest cost of energy, wins the bid and will be constructed in the area. In order to win tenders, Vattenfall is therefore motivated to reduce LEC for its projects.

The design of an offshore wind park includes different steps, such as turbine and components selection, the definition of the specific position of each turbine in the park, the (position-specific) design of the turbine foundations, the design of the cable routes to transfer the energy from the turbines to shore, the maintenance strategy, and so on. Before our project, the design of wind farms was usually done manually, in a sequential way. Figure 1.4 shows how the turbine allocation and the cable routes where designed when we started our collaboration with Vattenfall.

Figure 1.4: How a wind park was traditionally designed. Our work helped to improve the process, by including multiple steps directly in the optimization and by leading to better solutions



First the turbine position is drafted by the Wind-and-Site team, looking mainly at power production. This process is carried out partly manually and partly using commercial software. As we will see in Chapter 9, however, commercial software lacks at capturing some important factors in positioning the turbines, and typically generates suboptimal layouts.

The preliminary layout was manually changed to take additional constraints into consideration, not captured by the design software.

The layout was then passed on to the foundation team, that checked for costs and feasibility to actually locate the turbines in the given positions, given the specific soil conditions of the area. If infeasible or very expensive locations were identified, the layout was returned to the Wind-and-Site team for modifications. The design of each turbine foundation was carried out manually.

The layout was then given to the electrical team, which manually defined the cable connections between turbines, often just with pen and paper.

Constraints on the cable routings and additional cost factors (such as power losses in the cables) were checked in a second stage, and the original cable routing was possibly (manually) changed accordingly.

Each of these steps was iterated multiple times, before getting to a final (suboptimal) layout that satisfies all the real-world constraints. The whole process required considerable time and involved several people, and often resulted in a suboptimal layout.

In this PhD thesis we developed decision support tools for supporting the process. More specifically, we developed three optimizers: one for turbine allocation, one for cable routing, and one to help in the foundation design (Figure 1.5). In order to incorporate all the main constraints arising in practical applications, we closely collaborated with different teams in the company. Our final optimization models incorporate Wind-and-Site factors (maximizing the power production of the park), geographical constraints (considering, for example, obstacles in the site), civil engineering expertise (in the foundation costs), electrical knowledge (in the cable routing constraints/cost factors). The resulting optimization models are not only able to capture more constraints than any commercial software, but also to significantly outperform them in solution quality. Increasing park production and reducing costs lead to more competitive bids for site acquisition, and so a higher probability of winning tenders and of developing new parks. In addition, the time needed by the employees is now limited to the input definition, while most of the constraints in the final

layout are handled automatically by the optimizer.

In addition, our optimization tools allow the company to perform different scenario evaluations, being able to quantify the impact of decision choices already at design phase. This is very important when designing a new wind park or when evaluating new technologies on the market.

Figure 1.5: An offshore wind park (DanTysk). At design phase the turbine position must be selected, together with their foundation type. All the turbines must be connected to the offshore substation (yellow in figure) using cables. Source: Vattenfall, 2017

From an OR perspective, the motivation for this work is twofold. From one side, we are introducing and formalizing interesting optimization problems. The family of problems here considered, indeed, has so-far received limited attention from the OR community, and this thesis provides a link between the engineering and OR perspectives. We have developed new solution methods and, thanks to our collaboration with Vattenfall, we have acquired realistic data to test them and to prove their impact in real cases. On the other side, the problems considered involve a very large number of variables and non-linearities. How we handled these challenges can hopefully be of interest for other applications, where similar problems are faced.

More generally, being able to reduce LEC for offshore projects leads also to cheaper renewable energy on the market, and therefore potentially increases

the use of these clean sources over fossil fuels.

1.2 Problem definition

Offshore wind-park design optimization includes all the planning tasks to be performed at design phase. This phase is characterized by a large degree of freedom, which means large room for improvement and for evaluating different design options. In this framework we focused mainly on two specific areas in the design phase: turbine allocation (also denoted as wind farm layout) and inter-array cable routing. We briefly discuss also the optimization of a specific foundation type (i.e., jacket foundations). We here give a very brief definition of all three problems, while we refer the reader to Chapter 2 to a more detailed overview of the problems.

1.2.1 The wind farm layout optimization problem

The Wind Farm Layout Optimization (WFLO) problem can be defined as follows:

Given a specific site, its resource maps and terrain constraints,
determine a feasible allocation of turbines that maximizes power production,
subject to real-world constraints.

A key aspect in the optimization is to take wake effect into account. The wake effect is the interference phenomenon for which, if two turbines are located one close to another, the upwind one creates a shadow on the one behind. This is of great importance in the design of the layout since it results in a loss of power production for the turbines downstream, that are also subject to a possibly strong turbulence. Figure 1.6, that refers to the London Arrays' offshore wind farm, well illustrates the problem.

Figure 1.6: Wake effect on London Array wind farm. Photo by Thomas Hjort – all rights reserved

As we can see in the figure, under this wind scenario only the first turbine line is hit by full wind, while all the other turbines suffer a strong wake effect reducing production and increasing stress on turbines. A layout with strong wake effect should thus be penalized in our search of the optimal layout. As a result, one needs to model interference between turbines taking several possible wind scenarios (obtained from real-world samples) into account.

Note that wake effect can be generated not only by the turbines in the layout, but also by nearby parks.

Multiple constraints must be considered in the optimization, such as a minimum distance between turbines, and minimum/maximum number of turbines to locate. Limitations related to the seabed should also be considered, which translate in extra costs to construct turbines or in the definition of forbidden areas.

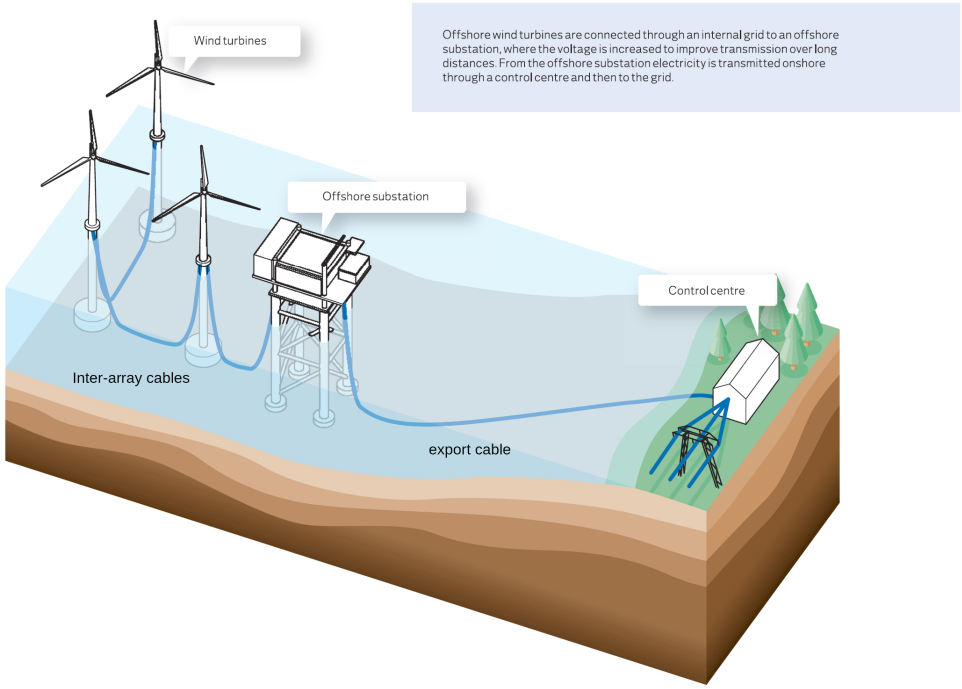
1.2.2 The inter-array cable routing optimization problem

The Offshore Wind Farm Cable Routing (OWFCR) problem concerns the optimization of the inter-array cable routing and can be defined as:

Given a turbine layout and a cable set, find a feasible cable connection between all turbines and the given substation(s), minimizing the total cable costs.

The main topic here is that the power production of the offshore turbines needs to be collected in substations to be transferred to the coast. To do that, each turbine must be connected to one substation through a cable path (cables connecting turbines are called inter-array cables, see Figure 1.7).

Figure 1.7: The inter-array cable routing is used to collect the energy from offshore turbines in one (or more) offshore substation(s), while the export cable transfers this energy to shore. Inspired from: Vattenfall, 2017



The optimization is subject to several technical requirements. The main ones are: the capacity of the selected cable, the capacity of the given substations (i.e., the maximum number of cables that can be connected to each substation), and the fact that cables should not cross. Cable crossing is not impossible in practice but is very expensive and increases the risk of damage of the cables,

and is therefore avoided by practitioners. The cable costs considered in our optimization are both the immediate cable costs, and the long-term revenue losses related to power losses in the cables. When the energy passes through each cable, indeed, part of it gets lost due to the electrical resistance of the cable. Different cables have different resistances, so a proper selection of the cables can result in considerable savings in the long term.

Different variants of the OWFCR problem can be defined. Due to our close collaboration with Vattenfall, and continuous feedbacks from practitioners on our results, we have been able to identify and solve other variants of the OWFCR problem of interest in practice. It can be relevant, for example, to impose different cable topologies to the layout (string structure, loop structure, etc.), to cope with extra costs for connecting multiple cables to a single turbine, or to handle possible cable failures. New emerging technologies can also be considered, such as Offshore Transformer Modules (Siemens, n.d.). We describe these extensions in detail in Chapter 8.

1.2.3 The jacket foundation optimization problem

At the end of the PhD project we also started to investigate another area of the offshore wind park design, namely the turbine foundation optimization. Different kinds of foundations exist (such as monopiles, gravity-based, tripod, etc.). Each foundation type is used for a specific water depth, or for specific soil conditions. In our work we looked into the optimization of tube selections within a complex structure (the so-called, jacket foundation, used for high water depths; see Figures 1.8 and 1.9). The problem can be stated as follows:

Given a specific jacket foundation structure and a list of possible tubes, determine a feasible tube selection that minimize the total structure weight, without jeopardizing the integrity of the structure.

A reduced total weight translates into a reduced cost for the overall structure. The optimization aims at reducing costs, while fulfilling all the necessary constraints on the structural integrity (damage level).

Figure 1.8: Steel jacket foundations. Source: Vattenfall, 2017

Figure 1.9: Installing a jacket foundation. Source: Vattenfall, 2017

1.3 Contribution

In order to reduce energy prices and to be more competitive on the market, every step of the design of a park should be properly optimized. In this thesis we developed different models and Mixed Integer Linear Programming (MILP) based algorithms that showed to significantly improve some of these steps. Six journal paper resulted from our work. Their main contributions are the following (in order of appearance in the thesis):

- Fischetti and Pisinger, 2018a gives a general overview of the problem arising in designing a wind park, and on how to solve them using OR techniques. It models, for the first time, cost of foundations in the optimization of the wind park layout. As an original contribution, a mathematical model for optimizing the tube selection of jacket foundations is presented and tested.
- Fischetti and Monaci, 2016 considers the optimal allocation of turbines subject to interference conditions. Our goal was the design of a fast heuristic capable of handling instances with 10 000+ potential positions in a matter of minutes. To this end, we have developed two strategies: a fast ad-hoc heuristic, and a MILP model designed for the very large instances of interest. A synergic use of these two tools has been proposed, following a clever MILP-and-refine recipe where two different variants of the underlying MILP model have been solved through a proximity search heuristic. Computational results on medium-to-large instances show that this approach outperforms a standard use of the two basic tools.
- Fischetti and Pisinger, 2017c introduces a new MILP model for optimal cable routing in offshore wind farms. A main novelty of this model is its capability of taking both installation costs and power losses into full account. A new matheuristic framework is also developed for difficult real cases. We have been able to describe the problem as it appears in real applications, and to validate our results on actual wind farms.
- Fischetti and Pisinger, 2018 (to appear) illustrates the impact of using a sound optimization to consider electrical losses in the design of offshore cable routes. To the best of our knowledge, this is the first detailed collection of what-if analyses on real-world intances. Such what-if analyses would not be possible without a sound optimization tool.
- Fischetti and Pisinger, 2018b introduces a set of new extensions to the basic offshore wind farm cable routing, not yet studied by the OR community. The paper presents four original model extensions, to consider

new engineering requirements or new technologies on the market. Real-world instances are used to evaluate the impact of these restrictions/new technologies in practice.

- Fischetti and Fraccaro, 2018 proposes a new way to merge Mathematical Optimization (MO) and Machine Learning (ML), where the optimization model comes first, and its optimized solutions are used as training set for a ML algorithm aimed at quickly estimating the value of optimized solutions. The paper proves that ML techniques can well estimate the optimal value for optimized wind farm layout instances.

Our wind farm layout optimization framework, was also used as an example in two book chapters, related to matheuristic techniques in general, namely
Fischetti and Fischetti, 2016
Fischetti et al., 2016

A combined version of the two is proposed in Chapter 3.

Reduced versions of the papers collected in this thesis appeared also as conference papers:
A Mixed-Integer Linear Programming approach to wind farm layout and inter-array cable routing, American Control Conference (ACC), 2015 (Fischetti et al., 2015a)
Inter-array Cable Routing Optimization Considering Power Losses, 14th Wind Integration Workshop 2015 (Fischetti et al., 2015b)
Inter-array cable routing optimization for big wind parks with obstacles, European Control Conference (ECC), 2016 (Fischetti and Pisinger, 2016)
On the Impact of using Mixed Integer Programming Techniques on Real-world Offshore Wind Parks, International Conference on Operations Research and Enterprise Systems (ICORES), 2017 (Fischetti and Pisinger, 2017b)
Mixed Integer Linear Programming for new trends in wind farm cable routing, International Network Optimization Conference (INOC), 2017 Fischetti and Pisinger, 2017a
Using OR+ AI to predict the optimal production of offshore wind parks: a preliminary study, Optimization and Decision Science (ODS) 2017, (Fischetti and Fraccaro, 2017)

1.4 Thesis outline

This thesis is organized as a collection of journal papers, with some additional chapters with extra information and examples. The overview paper Fischetti and Pisinger, 2018a (in Chapter 2) is used as an introduction to the thesis project, and it includes a literature review, an overview of our work and future work. The present chapter is therefore only used to motivate the thesis and to guide the reader through its chapters.

The thesis is divided into different parts, each focusing on a different aspect of wind farm design.

Part I is an introduction to the wind farm design problems considered and the mathematical optimization techniques used. Part I consists of:

- Chapter 1, used as a thesis guideline
- Chapter 2, based on the overview paper Fischetti and Pisinger, 2018a and used as a full introduction to the thesis
- Chapter 3, giving a deeper introduction to the Matheuristic techniques that are used in the thesis. This chapter is extracted from two book chapters: Fischetti and Fischetti, 2016 and Fischetti et al., 2016

Part II considers one of the two main optimization problems investigated in this thesis: the Offshore Wind Farm Layout (OWFL) problem. As already stated, this problem consists in the optimal allocation of turbines minimizing wake effect, subject to different constraints. In particular:

- Chapter 4 reports our main paper on the topic (Fischetti and Monaci, 2016). The optimization problem is explained in details together with the resolution method used.
- Chapter 5 studies possible extensions of the OWFL model, arising in practical applications. In particular, this work concerns the possibility of considering multiple turbine types in one area or different sites simultaneously.

Part III exploits the other main optimization problems of this thesis, namely the Offshore Inter-Array Cable Routing problem, i.e. to optimally connecting all the offshore turbines to substation(s) through cables. The chapters are:

- Chapter 6, reporting our main paper on the topic (Fischetti and Pisinger, 2017c);
- Chapter 7, containing the paper Fischetti and Pisinger, 2018 (to appear) where we focus on the industrial application of our Offshore Inter-Array Cable Routing model;
- Chapter 8, containing the paper Fischetti and Pisinger, 2018b where we focus on different extensions of our Offshore Inter-Array Cable Routing model.

Note that, in Part III, we have a main paper (Chapter 6) and its extensions (i.e. Chapters 7 and 8). In the papers regarding extensions of the model, introduction and basic model from the original paper are repeated and can eventually be skipped by the reader. We decided to include the full versions of the papers, so as to preserve their original structure and to make every chapter self-contained. To guide the reader, we provide reading instructions for each paper.

Part IV reports on our experience in Vattenfall and how our optimizers are today used inside the company. Our work, indeed, was developed as an industrial PhD project, so in very close contact with our industrial partner, who proposed us the optimization problems at first, and helped us to develop and test the resulting optimizers. Two chapters are presented:

- Chapter 9 illustrating how we tested our wind farm layout optimizer on a real-world case;
- Chapter 10 showing how our two optimizers (wind farm layout and cable routing) are now used in the business case definition for real projects, using the Danish Kriegers Falk wind park as an example.

Part V includes some Machine Learning (ML) applications. In particular:

- Chapter 11 reports our paper (Fischetti and Fraccaro, 2018) on the usage of Machine Learning and Operations Research to predict the production of optimized layouts.

Part VI is dedicated to final remarks. It consists of one chapter:

- Chapter 12 drives conclusions and outline some future work.

Appendix A gives more details on how we computed interference between turbines.

References

Fischetti, M. (2014). "Mixed-Integer Models and Algorithms for Wind Farm Layout Optimization". http://tesi.cab.unipd.it/45458/1/tesi_Fischetti.pdf. MA thesis. University of Padova.

Fischetti, M. and M. Fischetti (2016). "Matheuristics". In: *Handbook of Heuristics*. Ed. by R. Martí, P. Panos, and M. G. Resende. Springer International Publishing, pp. 1–33. ISBN: 978-3-319-07153-4. DOI: 10.1007/978-3-319-07153-4_14-1. URL: http://dx.doi.org/10.1007/978-3-319-07153-4_14-1.

Fischetti, M., M. Fischetti, and M. Monaci (2016). "Optimal Turbine Allocation for Offshore and Onshore Wind Farms". In: *Optimization in the Real World: Toward Solving Real-World Optimization Problems*. Ed. by K. Fujisawa, Y. Shinano, and H. Waki. Tokyo: Springer Japan, pp. 55–78. ISBN: 978-4-431-55420-2. DOI: 10.1007/978-4-431-55420-2_4. URL: https://doi.org/10.1007/978-4-431-55420-2_4.

Fischetti, M. and M. Fraccaro (2017). "Using OR+ AI to predict the optimal production of offshore wind parks: a preliminary study". In: *Optimization and Decision Science: Methodologies and Applications: ODS*. Springer, pp. 203–211.

Fischetti, M. and M. Fraccaro (2018). "Machine Learning meets Mathematical Optimization to predict the optimal production of offshore wind parks". In: *Computers and Operations Reserach*. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2018.04.006.

Fischetti, M., J. Leth, and A. B. Borchersen (2015a). "A Mixed-Integer Linear Programming approach to wind farm layout and inter-array cable routing". In: *American Control Conference (ACC), 2015*. IEEE, pp. 5907–5912.

Fischetti, M. and M. Monaci (2016). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: 10.1007/s10732-015-9283-4. URL: https://doi.org/10.1007/s10732-015-9283-4.

Fischetti, M. and D. Pisinger (2018 (to appear)). "On the impact of power losses in offshore wind farm cable routing". In: *Invited paper to the Springer book of ICORES 2017*.

Fischetti, M. and D. Pisinger (2016). "Inter-array cable routing optimization for big wind parks with obstacles". In: *European Control Conference (ECC)*. IEEE, pp. 617–622.

Fischetti, M. and D. Pisinger (2017a). "Mixed Integer Linear Programming for new trends in wind farm cable routing". In: *Electronic Notes in Discrete Mathematics, INOC*. Elsevier.

Fischetti, M. and D. Pisinger (2017b). "On the Impact of using Mixed Integer Programming Techniques on Real-world Offshore Wind Parks." In: *ICORES*, pp. 108–118.

Fischetti, M. and D. Pisinger (2017c). "Optimizing wind farm cable routing considering power losses". In: *European Journal of Operational Research*. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.07.061. URL: http://www.sciencedirect.com/science/article/pii/S037722171730704X.

Fischetti, M. and D. Pisinger (2018a). "Mathematical Optimization for offshore wind farm design: an overview". In: *Business and Information Systems Engineering*. DOI: 10.1007/s12599-018-0538-0.

Fischetti, M. and D. Pisinger (2018b). "Optimal wind farm cable routing: modeling branches and offshore transformer modules". In: *Network*. DOI: https://doi.org/10.1002/net.21804.

Fischetti, M., D. Pisinger, and I. Vranceanu (2015b). "Inter-array Cable Routing Optimization Considering Power Losses". In: *14th Wind Integration Workshop*.

Government of The Netherlands (July 2016). *Netherlands Offshore Wind Farm Borssele cheapest world wide*. URL: https://www.government.nl/latest/news/2016/07/05/netherlands-offshore-wind-farm-borssele-cheapest-world-wide.

Siemens (n.d.). *New AC Grid Access Solution from Siemens: Lighter, faster, cheaper*. http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2015/energymanagement/pr2015030151emen.htm.

Vattenfall (2017). *internal image storage*.

Weston, D. (December 2017). *Vattenfall bids in no-subsidy Duch tender*. URL: https://www.windpoweroffshore.com/article/1452992/vattenfall-bids-no-subsidy-duch-tender.

Weston, D. (November 2016). *Vattenfall wins Kriegers Flak with euro 49.9/MWh bid*. URL: https://www.windpoweroffshore.com/article/1414947/vattenfall-wins-kriegers-flak-%5C%E2%5C%82%5C%AC499-mwh-bid.

Chapter 2

Mathematical Optimization for offshore wind farm design: an overview

Martina Fischetti[a] · David Pisinger[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]Management Science, Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

Publication Status: Published as Fischetti and Pisinger, 2018

Reading Instructions: Overview of all the thesis work. To be considered as an introduction to the thesis.

Abstract: Wind energy is a fast evolving field, that has attracted a lot of attention and investments in the last decades. Being an increasingly competitive market, it is very important to minimize establishment costs and increase production profits already at the design phase of new wind parks. This paper is based on many years of collaboration with Vattenfall, a leading wind energy developer and wind power operator, and aims at giving an overview of our experience of using Mathematical Optimization in the field. We will illustrate some of the practical needs defined by energy companies, showing how optimization can help the designers to increase production and reduce costs in the design of offshore parks. In particular, we give an overview of the individual phases of designing an offshore wind farm, and some of the optimization problems involved. Finally we go in depth with three of the most important optimization tasks: turbine location, electrical cable routing and foundation optimization. The paper is concluded with a discussion of future challenges.

Keywords: Offshore wind farm design, Mathematical Optimization, Mixed Integer Linear Programming, Heuristics, Cable routing, Wind Farm Layout, Jacket structure optimization

2.1 Introduction

Environmental sustainability asks for a considerable reduction in the use of fossil fuels, looking to alternative sources of energy. As a consequence, increasingly more energy companies are investing, for example, in wind energy, creating a more competitive market for renewable energy. Particular attention is given to offshore solutions (wind parks located at sea). In this paper we will give a detailed overview of how the offshore wind park design is carried out in wind-energy companies, focusing on how Mathematical Optimization techniques can make an impact in reducing costs and increase production. We will mainly address the optimization tasks related to the design phase of a wind park. This is the initial phase in defining a new wind park, so there is more room for optimization. In particular, we focus on three specific problems — wind turbines location, connection of offshore turbines with cables and turbine foundation design — as these are some of the optimization tasks having the greatest impact.

2.1.1 Wind park design phases

In this overview paper we will focus on the design phase of offshore wind parks. Designing a wind park is a complex project, involving different expertises and a large number of optimization tasks. Most of the main optimization tasks of the problem are still not totally automated and commercial software ignores several important constraints. Generally speaking, the main steps in the design of a wind parks consist of:

- *site selection* – often decided by the government and put on tender;
- *data collection* – most of it is performed previous to the tender;
- *technology selection* – which includes, for example, selecting the manufacturer and the model for all the components of a wind park (e.g. selecting which turbines to consider for the park);
- *definition of the layout* – deciding where to locate the turbines in the site;
- *evaluation of foundation costs and soil conditions*;
- *cable routing* – deciding how to connect the turbines to the substation(s);
- *electrical studies* – defining the detailed electrical design, dimensioning equipment, computing power losses, proving compliance to grid codes, voltage levels and frequency limits in the connection to the grid;
- *design of each specific foundation* (for each selected location).

According to our experience, the design of a wind park is structured as follows. When a company decides to enter a tender to construct a new wind park in Europe, it generally receives an area (selected by the government) and GIS information about it, e.g. the wind statistics measured on the site, the seabed conditions, possible obstacles in the site, etc. The company can decide what turbine type to build in the site and where to locate the turbines within the boundaries of the given area. The total Megawatt (MW) production of the site is also given at tender phase, as the grid operator needs to ensure stability when the new park production is injected in the existing power grid system. Since only one type of turbine is built in each site (mainly for maintenance reasons), this MW restriction easily translates into a fixed number of turbines that can be built in the site. With all this information at hand, the first task that the company engineers normally face is to decide where to locate the turbines (i.e. the *wind farm layout optimization problem*). This is a very challenging task due

to the so-called *wake effect*. The wake effect is the interference phenomenon for which, if two turbines are located one close to another, the upwind one creates a shadow on the downstream turbine (see Figure 2.1). This is of great importance in the design of the layout since it results in a loss of power production for the turbine downstream, that is also subject to a possibly strong turbulence. It is estimated in Barthelmie et al., 2009 that, in large offshore wind farms, the average power loss due to turbine wakes is around 10-20% of the total energy production. It is then obvious that power production can increase significantly if the park layout is designed so as to reduce the wake effect as much as possible. As we will see, Mathematical Optimization can play an important role at this stage.

Figure 2.1: Wake effect in an offshore wind park Source: Vattenfall, 2017

Once the turbine positions are decided, the layout is generally forwarded to the electrical team. Offshore turbines need to be connected to shore with cables. The turbines are connected with lower voltage cables to an offshore substation where all the energy is collected – this is the so called *inter-array cable connection*. A unique high-voltage cable (called *export cable*) is used to transport the energy from the substation to shore. The substation and export cable can be established by the same company that constructs the park or can be established before tendering. In this paper we assume the second scenario, so substation(s) and export cable are assumed to be fixed a-priori. The offshore inter-array cable routing problem consists of finding the minimum cost connection of all offshore turbines. Different types of cables, with different capacities, electrical resistances and prices, can be used. This optimization task is still carried out manually in many companies, leading to highly suboptimal cable routes. As we will see, MILP and ad-hoc heuristics can be used to solve the inter-array cable routing problem. Considering cable losses when designing the cable route is also very important. Due to the resistance in the

cables, indeed, some energy gets lost in the transmission to the substation. An optimized selection of the cable structure and the cable type, can reduce the amount of current losses over the lifetime of the park.

While the electrical team works on the cable routing, another team works on the *turbine foundations*. Once the turbine position is identified, the specific locations are checked for sea bed conditions. Depending on the environmental conditions at each position, the water depth and the turbine type selected, different foundations can be designed for each turbine. Currently, the most used foundation type is the monopile, which is the simplest foundation available on the market. When the water is very deep, more complex structures need to be used as, for example, jacket foundations. Different optimization tasks can be identified in the foundation design, especially looking at the component selection. Here, the main challenge is to ensure that the foundation will be able to stand the different forces acting on it, due to the turbine movements but also the sea conditions (waves, currents and so on).

2.1.2 Literature overview

In this subsection we give a literature study of different optimization problems that may arise in the establishment and operation of an offshore wind park. In the following sections we will then go in depth with some of the most important optimization tasks when establishing an offshore wind farm.

Probably one of the most studied optimization tasks in the wind park design is the wind farm layout problem. As we will see in further details later, this is a very challenging task due to the wake effect.

The *wind farm layout problem* was first formulated as an optimization model in the master thesis Fagerfjall, 2010. The objective is to position wind turbines taking into account wake effect and sound limitations for surrounding areas. The work of Turner et al., 2014 also develops a mathematical programming framework for the wind farm layout problem, focusing on the wake effect modelling. The resulting nonlinear optimization model is approximated both as a quadratic integer program and as a mixed-integer linear program. Only a limited number of wind scenarios are considered in the paper. The paper Zhang et al., 2014 focuses on better capturing the nonlinearities of the wake effect, proposing a constraint programming and a mixed integer programming version of the model. Decomposition techniques are used to improve solution complexity. A continuous approach to the wind farm layout problem has been used in Kwong et al., 2012 and Kusiak and Song, 2010. The continuous models are

highly non-convex and turn out to be intractable from a computational viewpoint when considering real-world cases, especially when considering obstacles in the site.

The models presented in this overview have their origin in the MILP formulation of Archer et al., 2011. In Fischetti and Monaci, 2015 the formulation was extended, paving the way for an easier stochastic version (taking different wind scenarios into account). This MILP model, with some ad-hoc heuristics, is able to solve large instances.

The next problem in the design phase of a wind park is the *cable routing* of offshore parks. This task consists in finding the optimal connection among offshore turbines and some collection points at sea, i.e. the so-called substations. Bauer and Lysgaard, 2015 proposed a model based on an open vehicle routing problem formulation. The model assumes that only one cable can enter a turbine, a condition that is seldom met in real-world cases. Different solution approaches were proposed in Berzan et al., 2011, where a divide-and-conquer heuristic and an integer programming model were presented and tested on small instances. Furthermore, Dutta and Overbye, 2011 presented a clustering heuristic for cable routing. Finally, matheuristic approaches have proven to be very valuable in real-world applications, especially when taking losses into account Fischetti and Pisinger, 2017b.

Another important set of problems in offshore wind farm optimization regards the *maintenance of offshore parks*. An offshore wind farm demands frequent maintenance to avoid breakdown and production losses. Maintenance requires expensive resources, such as vessels or helicopters, so it is important to use them effectively. Optimization of vessel routing and of maintenance scheduling was studied in Dai et al., 2015, while Gundegjerde et al., 2015 was focused on the optimization of the fleet size, proposing a stochastic three-stage programming model. Gutierrez-Alcoba et al., 2017 used bi-level optimization to cope with real-time requests. On the first (tactical) level, the fleet composition for a certain time horizon is decided, while on the second (operational) level, its operations schedule is optimized, given failures and actual weather conditions. Decomposition methods were instead used by Irawan et al., 2017 to find the optimal schedule for maintaining the turbines, the optimal routes for the crew transfer vessels, and the number of technicians required for each vessel. The routes take several constraints into account such as weather conditions, the availability of vessels, and the number of technicians available at the base.

Other optimization challenges concern the *structure of the turbine* itself. Wind turbines are, indeed, very expensive engineering systems subject to high loads.

Turbine towers, support structures and foundation systems can be optimized in order to reduce costs while ensuring no damages in the overall structure. Muskulus and Schafhirt, 2014 gave an overview of the topic and of the literature in the field. Oest et al., 2017 focused on the optimization of a specific foundation type, i.e. jacket foundations. Jacket foundations are one of the most complex/expensive structures, normally used at high water depth or at difficult soil conditions.

Finally, optimization of *energy storage* is getting still more attention in the wind energy sector. Due to variation in production and electricity prices, it can be beneficial to store the produced energy in order to sell it when production is lower and prices are higher. This helps stabilizing the grid, but can also increase the profit of wind farms. In Hou et al., 2017 it was investigated how to couple an offshore wind farm with hydrogen storage. The resulting non-linear optimization model was solved using sequential quadratic programming methods and particle swarm optimization. Another solution to the variability of wind power, is to use hybrid systems, i.e. to compensate the wind energy downtimes with other energy sources. One example, is to use solar energy: Sinha and Chandel, 2015 gave an overview of optimization methods for the integration of photovoltaic and wind energy: mostly hybrid techniques and metaheuristics have been used for this task.

As offshore wind farms are getting older, we will in the coming years see an increased need for *decommissioning* the farms. Not much work has been done on optimizing this phase. Topham and McMillan, 2017 gave an overview of the tasks involved. These tasks include removing wind turbines, foundations, substations and cables, as well as onshore installations. Interesting optimization problems to be considered in decommissioning could be planning of the individual phases as well as transportation planning. Hou et al., 2016 presented an optimization model for the decommissioning, in which the foundations are reused, but turbines are replaced with newer models. The problem was solved through particle swarm optimization.

2.1.3 Outline of the paper

In the next sections we go in depth with three of the most important optimization tasks being part of designing an offshore wind farm. These tasks are the turbine location, the electrical cable routing, and finally optimization of foundations. The problem formulations and solution methods are based on our experience in collaborating with a leading energy company in wind farm design. Section 2.2 is dedicated to wind farm layout optimization and illustrates how to

use Mathematical Optimization techniques to solve this challenging optimization task. Section 2.3 focuses on offshore wind farm cable routing optimization. A mathematical formulation of the problem is presented and matheuristics approaches are developed for solving the model. Several real-world examples are considered in Subsection 2.3.1. Finally in Subsection 2.3.2 we show how power losses can be handled in the optimization. Section 2.4 is dedicated to the optimization of jacket foundations and is an original contribution of this paper. In particular, Subsection 2.4.1 shows how to model the optimization task using MILP models, while Subsection 2.4.2 illustrates the potential of this optimization on a case study. Finally, Section 2.5 concludes the overview and Section 2.6 proposes directions for future research.

Sections 2.2 and 2.3 are mainly based on Fischetti and Monaci, 2015 and Fischetti and Pisinger, 2017b, with some extensions (e.g., considering cost of foundations in the layout optimization) and some additional real-world examples.

2.2 A Proximity Search heuristic for wind farm layout

In this section we will describe solution methods for the offshore wind farm layout problem. This problem consists in finding an optimal allocation of wind turbines in order to maximize power production, taking the wake effect into account. The building area (site) and its resource maps are given on input. The optimizer considers:

a) a minimum and maximum number of turbines that can be built;

b) a minimum separation distance between any pair of turbines to ensure that the blades do not physically clash;

c) the interference between installed turbines (wake effect).

This problem is very challenging, due to the large number of possible positions, that can exceed 20000 in real-world applications. Fischetti and Monaci, 2014 and Fischetti, 2014 underline the importance of having a suitable formulation of the MILP model and MILP-based heuristics on top of it, for such a large-size problem. In the following we will briefly summarize this work.

The available sea area to construct the wind farm can be discretized in a number of possible positions by over-imposing a regular grid. Let V denote the set of all possible positions for a turbine and let

- I_{ij} be the interference (loss of power) experienced by site j when a turbine is installed at site i, with $I_{jj} = 0$ for all $j \in V$; Jensen's model Jensen, 1983 can be used to compute the interference;
- P_i be the power that a turbine would produce if built (alone) at position i;
- $N_{\min}$ and $N_{\max}$ be the minimum and maximum number of turbines that can be built, respectively;
- D_{MIN} be the minimum distance between two turbines;
- $dist(i, j)$ be the distance between sites i and j.

In addition, let $G_I = (V, E_I)$ denote the incompatibility graph with

$$E_I = \{\{i, j\} : i, j \in V,\, dist(i, j) < D_{\min}, i \neq j\}$$

and let $n = |V|$ denote the total number of positions.

In Fischetti and Monaci, 2015, a binary variable is defined for each $i \in V$:

$$x_i = \begin{cases} 1 & \text{if a turbine is built at position } i \in V; \\ 0 & \text{otherwise} \end{cases}$$

The original quadratic objective function (to be maximized)

$$\sum_{i \in V} P_i x_i - \sum_{i \in V} \Big(\sum_{j \in V} I_{ij} x_j\Big)\, x_i \tag{2.1}$$

is restated as

$$\sum_{i \in V} (P_i x_i - w_i) \tag{2.2}$$

where the variable w_i is defined as

$$w_i = \Big(\sum_{j \in V} I_{ij} x_j\Big)\, x_i = \begin{cases} \sum_{j \in V} I_{ij} x_j & \text{if } x_i = 1; \\ 0 & \text{if } x_i = 0 \end{cases}$$

and denotes the total interference caused by site i. The model then reads

$$\begin{aligned}
\max \quad & z = \textstyle\sum_{i\in V}(P_i x_i - w_i) && && (2.3)\\
\text{s.t.} \quad & N_{\min} \le \textstyle\sum_{i\in V} x_i \le N_{\max} && && (2.4)\\
& x_i + x_j \le 1 && \{i,j\} \in E_I && (2.5)\\
& \textstyle\sum_{j\in V} I_{ij} x_j \le w_i + M_i(1-x_i) && i \in V && (2.6)\\
& x_i \in \{0,1\} && i \in V && (2.7)\\
& w_i \ge 0 && i \in V && (2.8)
\end{aligned}$$

The objective function (2.3) maximizes the total power production by taking interference losses into account. Constraints (2.4) impose a minimum and a maximum number of turbines that can be constructed in the site, while (2.5) ensure the minimum distance between turbines. Constraints (2.6) relate variables w_i with interference. A big-M term M_i is used to deactivate the constraint in case $x_i = 0$, namely

$$M_i = \sum_{\substack{j\in V\\ [i,j]\notin E_I}} I_{ij}.$$

Finally (2.7) and (2.8) define our binary and continuous variables, respectively.

As shown in details in Fischetti and Monaci, 2015, using a single index variable w_i allows this model to solve larger instances compared with equivalent two-index models in the literature (e.g., Archer et al., 2011 and Fagerfjall, 2010). Another strength of this formulation is the ability of easily dealing with different wind scenarios. Indeed, the definition of the turbine power P_i and of the interference I_{ij} depends on the wind scenario considered, that greatly varies in time. Using statistical data, one can collect a large number, say K, of wind scenarios k, each associated with $P_i^k, I_{i,j}^k$ and with arising probability π_k. Using that data, one can write a Stochastic Programming variant of the previous model where only the objective function needs to be modified as

$$z = \sum_{k=1}^{K} \pi_k \Big(\sum_{i\in V} P_i^k x_i - \sum_{i\in V}\sum_{j\in V} I_{ij}^k x_i x_j \Big) \tag{2.9}$$

while all constraints stay unchanged as they only involve "first-stage" variables x. It is therefore sufficient to define

$$P_i := \sum_{k=1}^{K} \pi_k P_i^k \qquad i \in V \tag{2.10}$$

$$I_{ij} := \sum_{k=1}^{K} \pi_k I_{ij}^k \qquad i,j \in V \tag{2.11}$$

to obtain the same model (2.1)–(2.8) as before. Therefore, using this MILP formulation together with Jensen's model for wake effect, one can easily address the realistic situation in which many wind scenarios are considered, just by using a suitable definition of the input data; this is not the case for more sophisticated wake effect models, that typically lead to really huge stochastic programming variants.

As shown in details in Fischetti, 2014, this model is not suitable for large real-world instances, and a heuristic framework must be built around it.

The authors showed that large-scale instances (around 20 000 possible positions to locate turbines) can be solved on a standard PC, using some ad-hoc heuristics and a MILP-based heuristic scheme called Proximity Search Fischetti and Monaci, 2015.

When facing large-size problem it is standard practice to "warm start" the MILP solver, using a first heuristic solution (let us call it $(\tilde{x}, \tilde{w})$) to initialize the incumbent of the solver. However, it is often seen in practice (see e.g. Fischetti and Lodi, 2011) that this strategy is unlikely to produce improved solutions within acceptable computing times if the underlying MILP model is very large and the formulation is weak—as it happens in our context. So, a different use of the MILP solver is suggested, which is used to "search a neighborhood" of the heuristic solution $(\tilde{x}, \tilde{w})$, as in the so-called "Proximity Search" method Fischetti and Monaci, 2014. In the wind farm context, some simple ad-hoc heuristics are used to generate a first solution and then the MILP solver is used as a black-box to improve this first solution $(\tilde{x}, \tilde{w})$ in stages. At each stage, an explicit cutoff constraint

$$\sum_{i \in V} (P_i x_i - w_i) \geq \sum_{i \in V} (P_i \tilde{x}_i - \tilde{w}_i) + \theta \tag{2.12}$$

is added to the original MIP, where $\theta > 0$ is a given tolerance that specifies the minimum improvement required. The objective function of the problem can then be replaced by a new "proximity function" (to be minimized):

$$\Delta(x, \tilde{x}) = \sum_{j \in V:\, \tilde{x}_j = 0} x_j + \sum_{j \in V:\, \tilde{x}_j = 1} (1 - x_j) \tag{2.13}$$

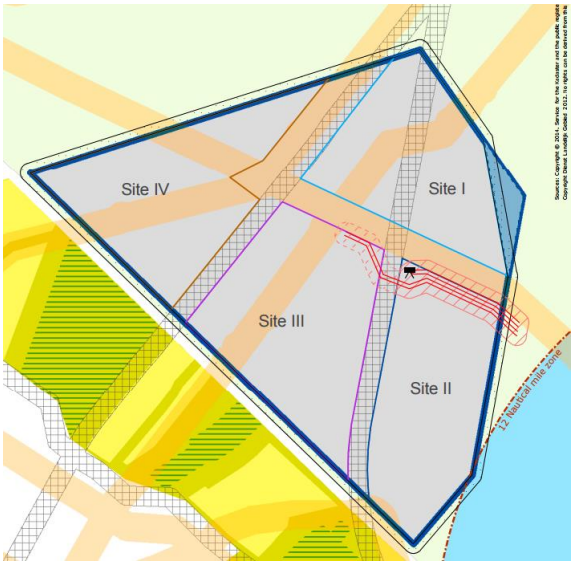
This function measures the Hamming distance between a generic binary vector x and the given $\tilde{x}$ (note that continuous variables w_i's play no role in this

definition). One then applies the MILP solver, as a black box, to the modified problem in the hope of finding an improved solution having a small Hamming distance from $\tilde{x}$. The computational experience reported in Fischetti and Monaci, 2014 confirms that this approach is quite successful. The proximity objective function is indeed beneficial both in speeding up the solution of the LP relaxations, and in driving the heuristics embedded in the MILP solvers. This method proved to be particularly valuable for the wind farm layout problem.

2.2.1 Real-world application

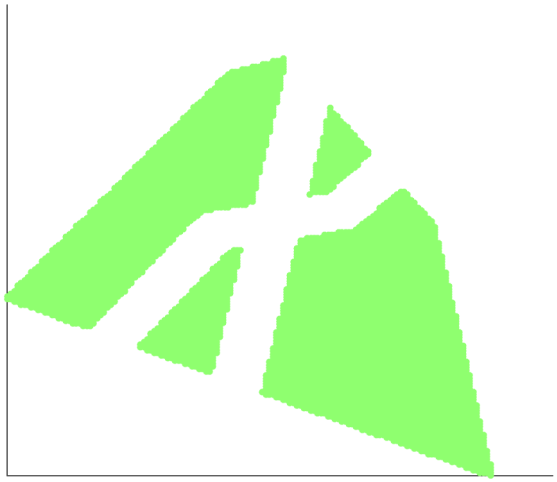
Using Mathematical Optimization techniques to optimize the turbine location can lead to huge savings. We used the optimization framework outlined in the previous section on a real wind park in The Netherlands. The Borssele area, in the Dutch province of Zeeland, was selected to construct a new wind park in 2016. The big offshore area was divided in 4 sites (Figure 2.2), and put on tender in two stages. In the first stage (summer 2016), sites I and II were on tender, for a combined 700-760 MW capacity. Here we will consider one of the two, namely Borssele I (350 MW capacity).

Figure 2.2: Borssele area, The Netherlands



The borders of the area were given at tender phase, as shown in Figure 2.2.

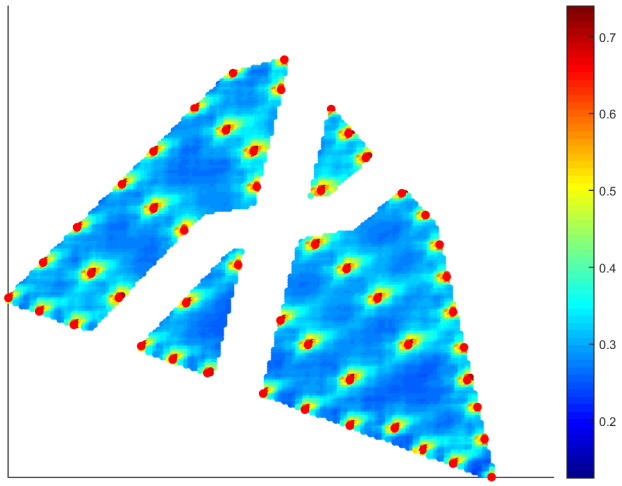
Figure 2.3: Borssele Site I



Part of the area was actually not available to construct turbines, due to pre-existing cables in the seabed: Figure 2.3 shows in green the area available to place turbines. It can be noticed that two corridors are forbidden. These kinds of "obstacles" are quite common in real sites, but they are easy to handle by our discrete model (by simply removing forbidden positions from the least of possible positions on input). In our experiment we were asked to locate 50 7MW turbines (154m rotor diameter) in the area. The company specified a minimum distance between turbines of 5 rotor diameters. 60,000+ wind scenarios were defined from real wind measurements in the site. The outcome of our optimization model is shown in Figure 2.4: the red dots represent built turbines while the colors on the background indicate interference.

We compared our result with the layout created using commercial software (see Figure 2.5). Our layout allows for an extra 0.57% Annual Energy Production (AEP) which, in the lifetime of a wind park, equates to more than 6 MEuros of extra income (net present value).

Figure 2.4: 50 Siemens 7MW-154 turbines at a minimum distance of 5 rotor diameters. Colors in the background represent interference over all the possible wind scenarios in input, considering their frequency.



2.2.2 Considering cost of foundations

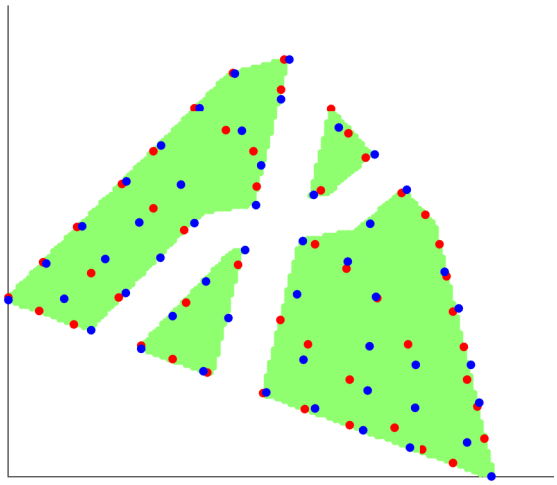
Costs of foundations can be a key factor, when the seabed conditions highly vary on the site. Due to waves, soil type and water depth, constructing a wind turbine in some positions of the site, could imply high extra costs. It is therefore valuable to include these costs in the optimization. To do so, we used the following strategy. Engineers from Vattenfall provided a cost map for the site: each possible position is associated with a construction cost, that was computed considering the foundation type, the weight of the turbine, the soil conditions for the specific position, and the water depth. Figure 2.6 shows the cost map for the site in hand (Borssele 1).

We slightly modified the objective function of model (2.1)–(2.8) as follows:

$$\max \quad \sum_{i \in V} [(P_i - \frac{c_i}{K_{euro}})x_i - w_i] \tag{2.14}$$

where c_i is the price of constructing a turbine in position $i'th$ (as specified on input) and K_{euro} is a factor to scale the price from €/KW to MW. To be specific, K_{euro} is the cost for each MW of production considering a park

Figure 2.5: Vattenfall layout (blue) vs Optimized layout (red)



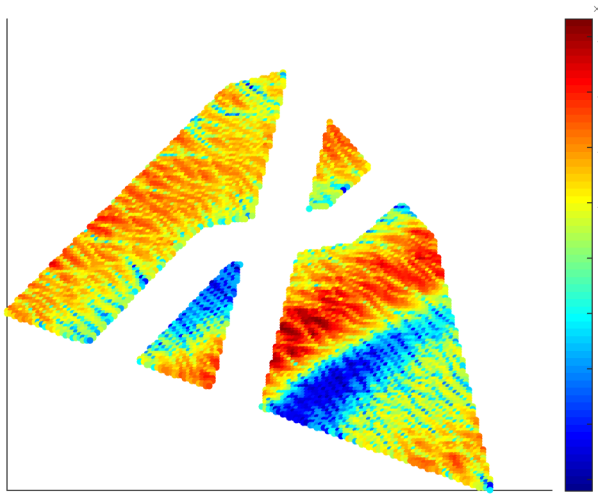
lifetime of 25 years and a Weighted Average Cost of Capital (WACC) of 8%. Both c_i and K_{euro} are problem specific and provided by Vattenfall.

Considering the same constraints as before and the same input data, but now including also foundation costs, we obtained the layout of Figure 2.7.

The cost of foundations was previously not considered by any commercial software used by Vattenfall. Therefore, the layout were usually defined based only on AEP, and eventually some turbines located in too expensive positions were manually moved, obtaining a suboptimal layout. Figure 2.8 shows a comparison between our layout and the one provided by the company. Company experts verified that our layout allows for an extra 0.28% production, while decreasing the cost of foundations of more than 10 M €. All in all, they estimated an increased income of more than 12 M € over the wind farm lifetime.

The Borssele example clearly shows the potential of using Mathematical Optimization techniques as an integrated part of designing offshore wind parks.

Figure 2.6: Foundation costs map for Borssele 1: different colors represent different costs to built a turbine in the specific position (the exact costs have been hidden due to privacy issues) .



2.3 Matheuristics for cable routing

We now assume that the turbine layout has been optimized and fixed, and we wish to find optimal cable connections between all turbines and the given collection point offshore (i.e. the substation(s)), minimizing the total cable costs. The optimization problem considers that:

- the energy leaving a turbine must be supported by a single cable;
- the maximum energy flow (when all the turbines produce their maximum) in each connection cannot exceed the capacity of the installed cable;
- different cables, with different capacities, costs and electrical resistances, can be installed;
- cable crossing must be avoided;
- a given maximum number of cables can be connected to each substation;

Figure 2.7: Optimized solution for Borssele I considering AEP and foundation costs. Black dots represent turbines, while background colors shows the foundation costs.

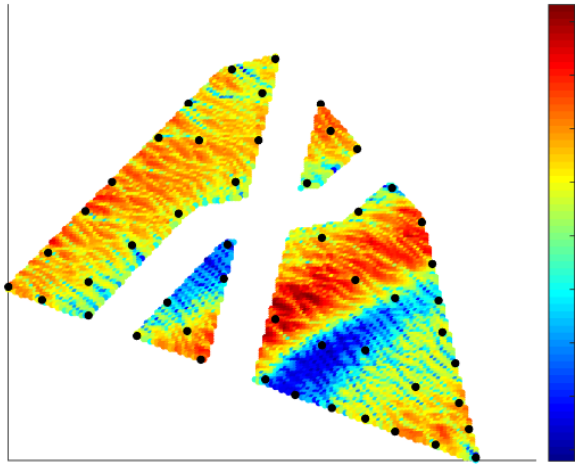
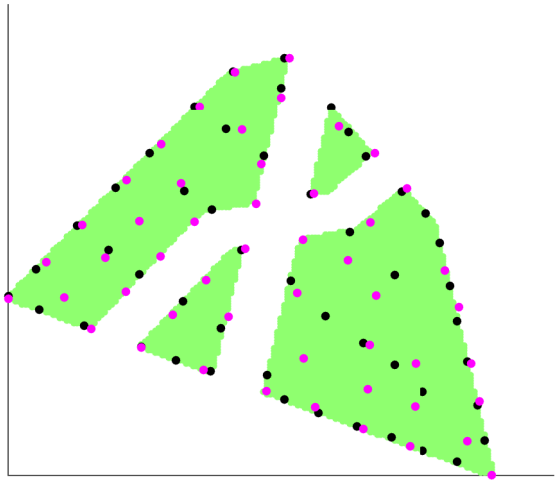


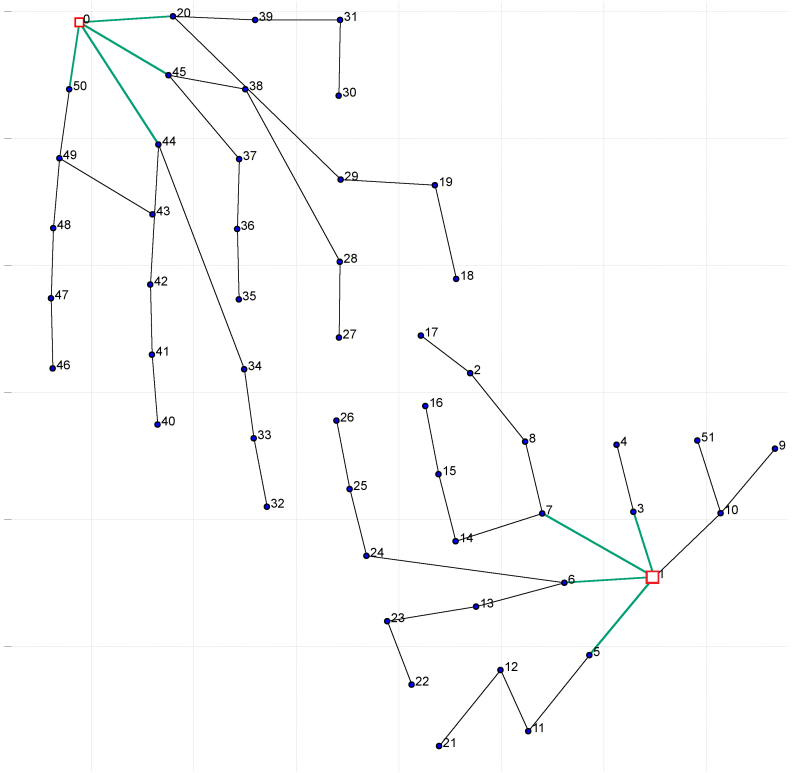
Figure 2.8: Optimized layout considering wake effect and costs of foundations (black) versus Vattenfall layout (pink)



- cable losses (dependent on the cable type, the cable length and the current flow through the cable) must be considered.

Figure 2.9 illustrates a possible cable routing.

Figure 2.9: An example of cable routing: all turbines (black dots) are connected with one of the two substations (red squares).



Following Fischetti and Pisinger, 2017b, we model turbine positions as nodes of a complete and loop-free directed graph $G = (V, A)$, and all possible cable connections between them as directed arcs. Some nodes correspond to the substations that are considered as the roots of the trees, being the only nodes that collect energy. Let P_h be the power production at node h. We distinguish between two different types of node: V_T is the set of turbine nodes, and V_0 is the set of substation nodes. Let T denote the set of different cable types that can be used. Each cable type $t \in T$ has a given capacity k_t and unit cost u_t, representing the cost per meter of cable – immediate costs, i.e. capital

expenditure (CAPEX). Arc costs can therefore be defined as $c^t_{i,j} = u_t dist(i,j)$ for each arc $(i,j) \in A$ and for each type $t \in T$, where $dist(i,j)$ is the Euclidean distance between turbine i and turbine j. The model uses the continuous variables $f_{i,j} \geq 0$ for the flow on arc (i,j). The binary variables $x^t_{i,j}$ define cable connections as

$$x^t_{i,j} = \begin{cases} 1 & \text{if arc } (i,j) \text{ with cable type } t \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, variables $y_{i,j}$ indicate whether turbines i and j are connected (with any type of cable). Note that variables $y_{i,j}$ are related to variables $x^t_{i,j}$ as $\sum_{t \in T} x^t_{i,j} = y_{i,j}$. The overall model can be stated as follows Fischetti and Pisinger, 2017b:

$$\begin{aligned}
\min \quad & \textstyle\sum_{i,j \in V} \sum_{t \in T} c^t_{i,j} x^t_{i,j} && && (2.15)\\
\text{s.t.} \quad & \textstyle\sum_{t \in T} x^t_{i,j} = y_{i,j}, && i,j \in V : j \neq i && (2.16)\\
& \textstyle\sum_{i : i \neq h} (f_{h,i} - f_{i,h}) = P_h, && h \in V_T && (2.17)\\
& \textstyle\sum_{t \in T} k_t x^t_{i,j} \geq f_{i,j}, && i,j \in V : j \neq i && (2.18)\\
& \textstyle\sum_{j : j \neq h} y_{h,j} = 1, && h \in V_T && (2.19)\\
& \textstyle\sum_{j : j \neq h} y_{h,j} = 0, && h \in V_0 && (2.20)\\
& \textstyle\sum_{i \neq h} y_{i,h} \leq C, && h \in V_0 && (2.21)\\
& x^t_{i,j} \in \{0,1\}, && i,j \in V, t \in T && (2.22)\\
& y_{i,j} \in \{0,1\}, && i,j \in V && (2.23)\\
& f_{i,j} \geq 0, && i,j \in V, j \neq i. && (2.24)
\end{aligned}$$

The objective function (2.15) minimizes the total cable layout cost. Constraints (2.16) impose that only one type of cable can be selected for each built arc, and defines the $y_{i,j}$ variables. Constraints (2.17) are flow conservation constraints: the energy (flow) exiting each node h is equal to the flow entering h, plus the power production of that node (except if the node is a substation). Constraints (2.18) ensure that the flow does not exceed the capacity of the installed cable, while constraints (2.19) and (2.20) impose that only one cable can exit a turbine and none can exit the substations (tree structure rooted at the substations). Finally, constraints (2.21) impose the maximum number of cables (C) that can enter each substation.

In order to model no-cross constraints we need a constraint for each pair of crossings arcs, i.e., a very large number of constraints. We have therefore

decided to generate them on the fly, as also suggested in Bauer and Lysgaard, 2015. In other words, the optimizer considers model (2.15) - (2.24) and adds the following new constraints whenever two established connections (i, j) and (h, k) cross

$$y_{i,j} + y_{j,i} + y_{h,k} + y_{k,h} \leq 1. \tag{2.25}$$

The reader is referred to Fischetti and Pisinger, 2017b for stronger versions of those constraints. Using this approach, the number of non-crossing constraints actually added to the model decreases considerably, making the model faster to solve. Again, also in this application, the size of the problem is a main issue. As presented, the model is able to deal with small instances only, failing to find even a first feasible solution for large-size real-world instances. In order to produce high quality solutions in an acceptable amount of time also for large-scale instances, a matheuristic framework (as the one proposed in Fischetti and Pisinger, 2017b) can be used on top of this basic model. The main ideas behind our matheuristic framework are the following.

First, as we have already discussed previously, we know that warm starting the MILP solver with an initial solution can boost the resolution of large-size problem. In this application in particular, we decided to generate the first feasible solution using the MIP solver itself but on a relaxed version of the model. In the relaxed version of the model we allow for disconnected solutions, highly penalizing them in the objective function. Standard MILP solvers used on the relaxed model can quickly find a first (often disconnected) solution.

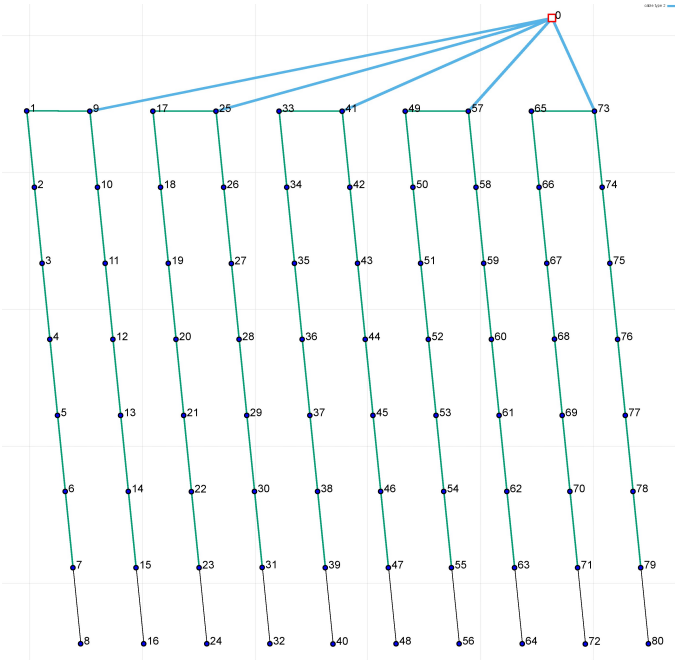
Secondly, we noticed that the difficulty of our problem was due to the large number of variables, i.e. the large number of possible cable connections in the complete directed graph. On the other hand, we also noticed that, once some arcs are fixed in the solution, the number of variables to optimize was highly reduced due to the no-cross constraint. From these observations, we designed the following hybridization of exact mathematical modeling and heuristics (i.e. matheuristic): we define a first feasible solution (x^*, y^*) using the MILP solver on the relaxed model, then we fix to 1 some of the y variables with $y^*_{i,j} = 1$. As said, fixing some arcs implies to exclude all the crossings arcs, with a drastic reduction in the dimension of the model. In order to decide which arcs to fix in the solution, we used different heuristic strategies, namely random fixing, string-based fixing, distance-to-substation fixing and fixing by sectors. For the sake of space we decided not to include more details on the heuristics here; the reader is referred to Fischetti and Pisinger, 2017b for more information. So, at each iteration, we temporarily fix to 1 some y variables and apply the preprocessing described above to temporarily fix some other y variables to zero. We then apply the MILP solver to the corresponding restricted problem, and we warm start the solver by providing the current solution (x^*, y^*). We abort the execution as soon as a better solution is found, or a short time limit of a

few seconds is reached. Then all fixed variables are unfixed, and the overall approach is repeated until a certain overall time limit (or maximum number of trials) is reached. Finally, the exact MILP solver is applied to the original model without any heuristic variable fixing, using the best-available solution to warm-start the solver.

2.3.1 Real-world application

As a practical illustration we consider the cable routing of the existing wind park of Horns Rev 1, a real-world offshore park located in Denmark. Figure 2.10 shows the actual design for Horns Rev 1 (from Kristoffersen and Christiansen, 2003).

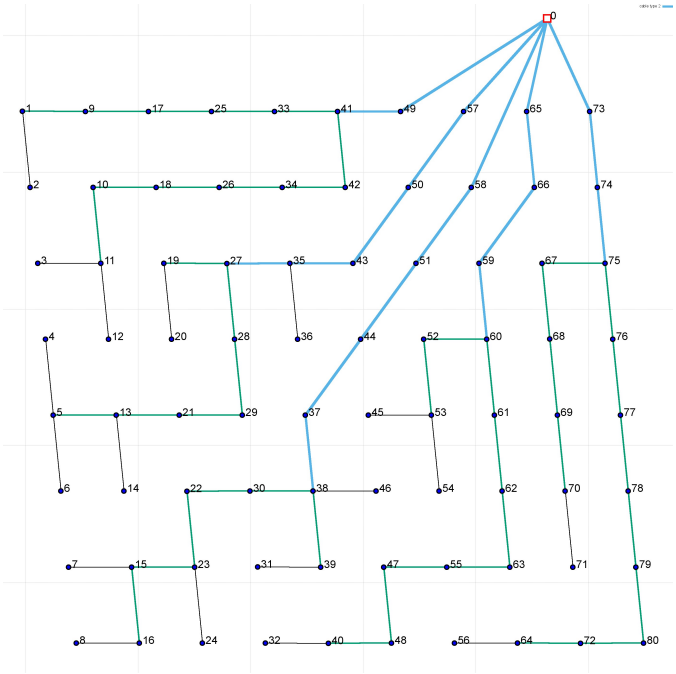
Figure 2.10: Existing cable routing for Horns Rev 1 Fischetti and Pisinger, 2017c.



Three different types of cables can be used: the thinnest cable supports one turbine only, the medium supports 8 turbines, and the thickest 16. Based on the cable cross section, we estimated the costs and resistances of these cables. The estimated prices are 85 €/m, 125 €/m and 240 €/m, respectively, plus

an estimated 260 €/m for installation costs (independent of the cable type). Using the matheuristic techniques of Section 2.3 on this case, we obtained the layout in Figure 2.11. The optimized layout is significantly different from the existing one: in terms of immediate costs, the optimized layout is more than 1.5 M € less expensive.

Figure 2.11: Optimized layout for Horns Rev 1 (CAPEX costs only): this layout is more than 1.5 M€ cheaper than the existing one Fischetti and Pisinger, 2017c.



2.3.2 Cable losses

When the energy passes through a cable, there is a loss due to the electrical resistance of the cable. Different types of cables with different electrical resistances are available on the market. Therefore, one should aim at minimizing not only the immediate costs (CAPEX) but also the future revenue losses due to power losses. This latter aspect is very important in practice, in that more expensive cables/layouts can be significantly more profitable in the long run. This issue is explicitly considered in Fischetti and Pisinger, 2017b, where a

precomputing strategy is developed to include the losses in the optimization without increasing too much the size of the model. The main idea is that the current loss on a cable can be computed by knowing the (discrete) number of turbines connected to that specific cable. Due to the limited capacity of the cables, the revenue loss due to cable losses for each possible combination of cable type and number of turbines connected can be precomputed. As a result, by just changing the input prices of the cables, one can consider revenue losses without any change in the MILP model; see Fischetti and Pisinger, 2017b for details. In Fischetti and Pisinger, 2017c the authors analyze the impact of considering cable losses in real-world instances. The results show that, in some cases, several hundred thousand euros can be saved in the long run for a single cable routing, when considering losses already in the design phase (compared with a layout optimized on immediate costs only). Here, we still use to the Horns Rev 1 example we introduced previously, to give an illustration of the potential savings with respect to a manual (existing) layout.

As we have already seen, without considering losses in the optimization, the optimized layout for Horns Rev 1 would look as in Figure 2.11. We can assume that the company decides to use this layout, making it possible to a-posteriori compute the losses related to it. It is still more profitable (by about 1.6 M€) than the existing one.

By optimizing cable losses, however, one can further improve its value in the long term. Figure 2.12 shows the optimized solution considering losses (thus optimizing the value of the cable route in its lifetime). Compared with the existing layout (Figure 2.10), this new layout is about 1.7 M€ (NPV) more profitable in 20 years, and still around 1.5 M€ cheaper at construction time.

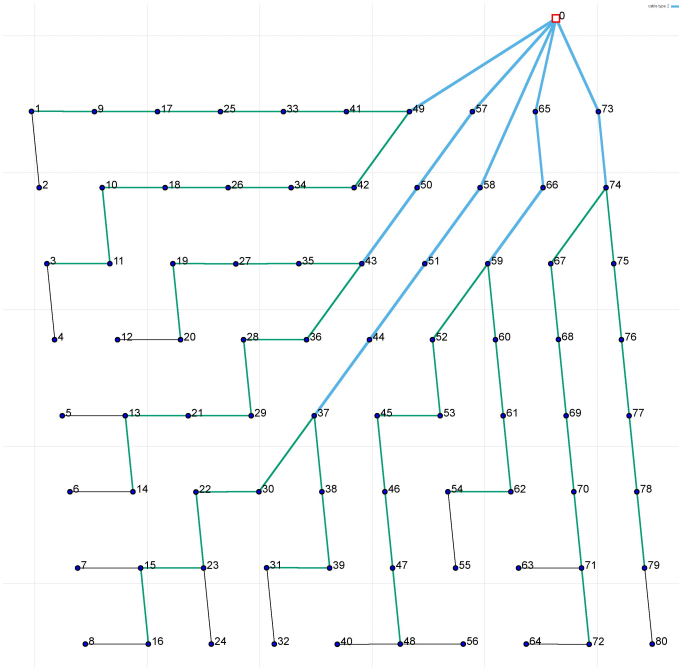
Table 2.1 summarizes the savings of the two optimized layouts compared with the existing one, both from an immediate cost perspective and from a long-term perspective.

Table 2.1: Savings of optimized solutions compared with the existing cable routing for Horns Rev 1 Fischetti and Pisinger, 2017c.

	Savings [M€]	
opt mode	immediate	in 25years
CAPEX	1.54	1.60
lifetime	1.51	1.68

The Horns Rev 1 example shows the impact of using Mathematical Optimiza-

Figure 2.12: Optimized layout for Horns Rev 1 (considering losses): in the 20-years wind park lifetime this layout is estimated to be more than 1.7 M €more profitable than the existing one Fischetti and Pisinger, 2017c.



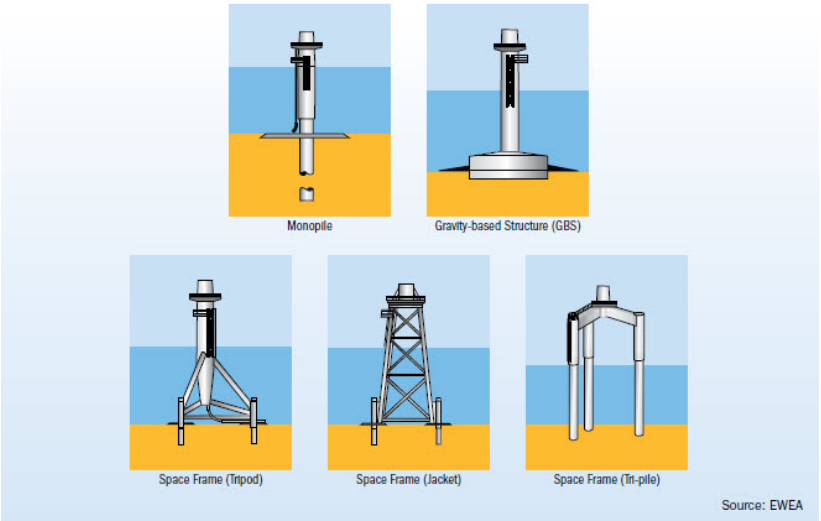
tion models for real world cable routing. The optimized layout is more than 1 M€ less expensive than the original (manual) one. Our model can be further extended to include additional constraints. Having a model where new constraints can easily be added is a key feature in a fast developing field of application as the wind-energy one. The authors showed in Fischetti and Pisinger, 2017a that this model can be extended to consider e.g. a maximum number of branches, loop structures to reduce the risk of cable failures and use of new technologies on the market. They also show the potential savings in considering (or not considering) these additional constraints in practice. Our optimization tool is able to solve real-world instances in a matter of minutes, allowing for different what-if analyses. Being able to quantify the impact of a design choice and to conduct a fast what-if analysis are key features for Vattenfall, all of which being impossible without a proper optimization tool.

2.4 Jacket Foundation Optimization

As wind farms are getting larger and more remotely located, installation and infrastructure costs are rising. In particular, offshore turbines are getting bigger and bigger, and heavier foundations are required.

Different foundation types exist, depending on the seabed conditions and on the turbine size; see Figure 2.13 for an illustration. In this section we focus on jacket (or space frame) foundations, that are one of the most complex/expensive structures, normally used at high water depth or for difficult soil conditions.

Figure 2.13: Different types of foundation — image from EWEA, 2012

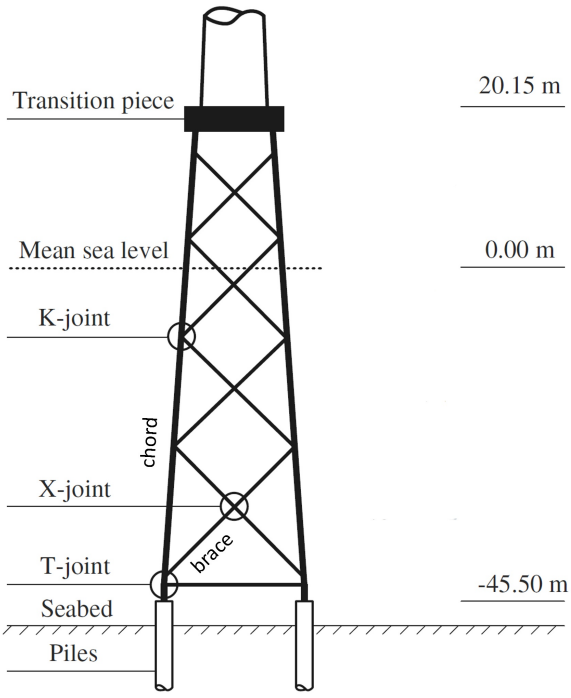


Once constructed, the foundation structures must resist stresses caused by the weight of the turbine, the wind that impacts it and the wave/currents in the sea area. More specifically, when designing a jacket structure, the designer has to choose a set of appropriate dimensions for the structural tubes for the space frame. The tube sizes, i.e. diameter and wall thickness, are chosen such that the joints can withstand the stresses that arise due to the loads. If the tube dimensions are chosen too slender, the stresses would exceed threshold values, leading to premature fatigue failure. On the other hand, if the tube dimensions are chosen too big, the tubes will be under-utilized and the overall structure will be too heavy and expensive. Thus, it is delicate balance to find the optimal selection of tube dimensions.

When optimizing it is therefore of crucial importance to consider these forces; for such a measure, we used the DNV-GL standards commonly used by practitioners DNV-GL, 2005.

A jacket foundation is identified by a structure, i.e., by a collection of *joints* and *tubes*. Tubes are called *chords* or *braces*. Chords are the main vertical columns carrying the overall loads and are normally of bigger dimensions. The horizontal and diagonal tubes are called braces and act as stiffeners. Their dimensions are usually smaller than the chords. A chord is connected with one or more braces by welding. Joints can be of different types, depending on how many tubes are connected through them. As all the faces of the jacket are identical, it is common practice to visualize the 3D structure as its 2D projection. Looking at a 2D representation of a structure, it is easy to identify the joint types: T-types connect a chord and a brace, K-types connect a chord and 2 braces, and X-types connect 2 braces. Figure 2.14 shows a 2D representation of a jacket foundation.

Figure 2.14: Basic components of a jacket foundation. Picture inspired by Oest et al., 2017



The company experts provided a list of possible tube types as input data: each of them has a specific diameter [mm], a specific thickness [mm] and unit mass [kg/m]. These tube types come from a standard list offered by the tube manufacturer. In principle, the company could require the manufacturers to design customized tubes, but this has an extra cost; this is why, the aim is to use only standard tubes, leading to great savings. The optimization task consists in optimally selecting these standard components, minimizing the total structure cost, while ensuring no premature fatigue failures.

2.4.1 The optimization model

Input data includes the shape of the jacket foundation to be built (i.e., the joints, chords and braces and the way they are connected) and a set of tube types T. Each tube type $t \in T$ has a different mass, say m_t. We aim at optimizing the tube selection in order to minimize the total mass of the structure, subject to the following requirements:

- the tube type should be able to handle the local stresses (damage constraint);
- chord tubes should have a larger diameter than brace tubes;
- only one tube type should be selected for each connection.

The problem is naturally formulated on a directed graph $G = (V, A)$ where the set of nodes V contains all the joints, and the set of arcs A contains all the tubes. We can then define a binary variable x_a^t for each $a \in A$ and $t \in T$, where $x_a^t = 1$ iff arc a has a tube of type t. Different forces will act on each arc. In particular, if two generic nodes i and j are connected through an arc a, this arc will cause different loads on i and j. To capture this in our model, we created a copy of all the given arcs, directing them so that we associated to $a = (i, j)$ the forces acting on j because of the a connection, while we associate to its symmetric arc $a' = (j, i)$ the forces acting on i because of a. Note that arc orientation is only conventional, in that only one tube actually exists in the jacket structure (so we impose that $x_a^t = x_{a'}^t$).

Our MILP model then reads

$$\min \quad \tfrac{1}{2} \sum_{a \in A} \sum_{t \in T} (l_a m_t) x_a^t \tag{2.26}$$

$$\text{s.t.} \quad \sum_{t\in T} x_a^t = 1 \qquad a \in A \tag{2.27}$$

$$x_a^t - x_{a_1}^t = 0, \qquad t \in T, \text{ symmetric arc pair } (a, a_1) \in A \times A \tag{2.28}$$

$$x_a^t + x_{a_1}^{t_1} \leq 1, \qquad \{a, a_1, t, t_1\} \in \overline{T}, \tag{2.29}$$
$$\{a, a_1\} \text{ in a T-joint, with } a \neq a_1,\ t, t_1 \in T$$

$$x_a^t + x_{a_1}^{t_1} + x_{a_2}^{t_2} \leq 2, \qquad \{a, a_1, a_2, t, t_1, t_2\} \in \overline{K}, \tag{2.30}$$
$$\{a, a_1, a_2\} \text{ in a K-joint, with } a \neq a_1 \neq a_2,\ t, t_1, t_2 \in T$$

$$x_a^t \in \{0, 1\}, \qquad a \in A, t \in T. \tag{2.31}$$

In the objective function (2.26) we minimize the total mass structure. Note that the contribution of each single arc a to the total mass of the structure is given by the unit mass of the specific tube type t selected (indicated as m_t) multiplied by the length of the arc (l_a). Constraints (2.27) impose that one type of tube is selected for each arc in the structure, while constraints (2.28) impose that the same tube type is used for symmetric arcs. Constraints (2.29) and (2.30) forbid infeasible tube connections. In particular, set $\overline{T}$ contains all the pairs of arcs connected in T-connections which are infeasible due to limits on the damage levels. Analogously, set $\overline{K}$ in (2.30) refers to K-joints: it contains all the arcs connected in a K-joint that are infeasible due to limits on the damage levels. Both sets $\overline{T}$ and $\overline{K}$ include also pair or triplets (respectively) where the braces are bigger than the chord (as required by Vattenfall's engineers). Finally (2.31) requires that all variables are binary. It can be noticed that the MILP model does not depend explicitly on the actual damage formulas, which are only used for the definition of the no good sets $\overline{T}$ and $\overline{K}$ (in our implementation, we used official standards in the field DNV-GL, 2005, but different formulas can be implemented as well).

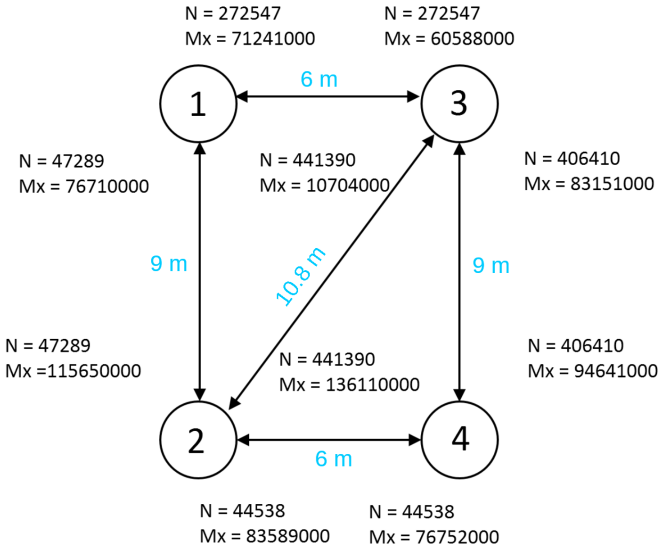
2.4.2 Preliminary results

For simplicity, we are applying the MILP model on a simple, but representative structure consisting of T and K joints. The structure is a simple structure, but the applied methodology is easily expandable to real life structures with more complex joints. As already mentioned, in a jacket foundation all the faces of the structure are identical, therefore the study is carried out on one of the faces, and then extended to the others. Therefore, we will next consider a 2D representation of our 3D jacket foundation example.

As already discussed in Section 2.4.1, a jacket foundation can be represented as a graph with joints as nodes and tubes as arcs. We will use this representation

for our test example. We are given from Vattenfall's experts the structure of the foundation (i.e., the set of nodes and arcs) and we have to determine the tube type to be used for each arc. We are also given the different forces acting on each arc of the structure. Forces can be different from one side of the tube to the other, therefore we are given forces for each extreme of any physical connection (as shown in Figure 2.15). Based on the sectional forces N and M_x in Figure 2.15, an expected fatigue lifetime of the joint is calculated based on a set of parametric joint formulas DNV-GL, 2005.

Figure 2.15: Our illustrative example. The figure shows the structure of one face of the jacket foundation, the length of the tubes, and the different forces acting on them. N is the axial force in Newtons and the moment Mx is the inplane bending moment in Newton-millimeters.



We are also given a set of possible tube types(as in Table 2.2) to use in each of the physical connections. Each tube type is characterized by its diameter, thickness and unit mass. The diameter and thickness of each tube impact its capacity of withstanding different forces. We aim at minimizing the total mass of the structure while ensuring that the structure can withstand the different forces acting on it.

We performed the tube type selection using our optimization model of Section 2.4.1, while an expert from Vattenfall performed the same task manually. The

Table 2.2: Possible tube types to use.

diameter [mm]	thickness [mm]	mass [kg/m]
400	30	278
610	30	429
610	40	562
610	50	691
711	30	504
711	36	599
711	45	739
813	30	579
813	40	763
813	50	941
1219	30	880
1219	40	1163
1219	50	1441
1422	30	1030
1422	40	1363
1422	50	1692
1626	30	1181
1626	40	1565
1626	50	1943

optimization solver (IBM ILOG Cplex 12.6) reached optimality in a matter of seconds. Table 2.3 shows a comparison with the manual solution. In the first column of Table 2.3 we report the arc, then we specify the tube selected manually (second and third column) or by the optimization model (fourth and fifth column). Finally, in the last table row, we compare the total mass of the two feasible solutions (both satisfying the damage constraints).

As it can be seen from the last row of Table 2.3, the optimized structure is much lighter than the manually constructed one (about 5 Tons less, with a saving of more than 12%). This is a very interesting result: although the toy structure was really simple, the optimization could still significantly outperform the manual approach. We therefore expect to have even larger savings for more complex (real-world) structures, where the manual task is much more difficult to carry out. Furthermore, by having an automated process, the designers would be more willing to perform additional design iterations, since they do not have to carry out the tedious optimization job manually for each design iteration.

Table 2.3: Solution of the tube selection optimization problem: the manual solution (left) vs the optimized one (right)

	Manual		Optimized	
connection	diameter	thickness	diameter	thickness
(1,2)	1219	50	1219	30
(3,4)	1219	50	1219	30
(1,3)	711	30	1219	30
(2,3)	813	40	1219	30
(2,4)	610	30	813	30
Total mass	39.8 Tons		34.1 Tons	

2.5 Conclusions

In order to make wind energy competitive with non-renewable energy sources, every part of an offshore wind farm must be optimized to improve efficiency and reduce costs. In this overview we have shown how Mathematical Optimization can significantly improve several steps of the design phase. In particular we have addressed turbine allocation, inter-array cable routing and optimization of jacket foundations. Given the large size and complexity of the instances, matheuristic techniques have been developed and used to optimize real-world wind farms. We have shown that millions of Euros can be saved in this way. Using MILP-based models, rather than manual solutions, we have also been able to quantify the impact of different design choices and to carry out different what-if analyses (for example considering power losses in cable routing, or considering cost of foundations in the wind park layout). This is extremely interesting from an application perspective, in that it allows the company to have a better understanding of the case in hand and to take informed decisions.

2.6 Future work

Still, many optimization challenges have not been solved in the wind field. Looking at the problems we considered in this overview, an interesting next step would be to look at the integration of the different optimization phases in wind park design. As we have seen, the wind farm layout model tends to spread turbines as much as possible, in order to reduce wake effect. On the other hand, the further apart the turbines are located, the higher becomes the infrastructure costs to connect them. It would therefore be interesting to

integrate, for example, the optimization of wind farm layout and cable routing together. Given the large size and complexity of both problems, some challenges would arise in solving a unified mode.

The models presented in this overview could also be generalized to onshore parks. Onshore wind park design is more complex, as it includes some additional constraints and non linearities. In particular, most of the work on wind farm layout optimization assumes the wind to blow uniformly in the site. This assumption does not hold in onshore sites, where the shape of the land (mountains, hills, forests, etc.) impacts the free-wind speed. Furthermore, some extra constraints must be taken into account in the onshore case, such as noise limitations for nearby houses, or road connections to the turbines.

Looking further ahead, the wind energy sector is quickly evolving, so new technologies have to be considered. An example could be the Offshore Transformer Modules (OTM) that just recently entered the market. Those transformers are meant to substitute offshore substations, and can be connected to the turbine directly. Each turbine equipped with an OTM, can be connected both to inter-array cables and to higher-voltage cables (i.e. export cables). Considering OTMs in the cable routing optimization opens up for some new and interesting optimization tasks, such as deciding the number of OTMs in a park, deciding their position, etc. The presented MILP solution framework can easily incorporate new constraints, therefore it is very suitable for such a fast evolving field. Finally, floating wind farms are slowly appearing. Optimizing the establishment and operation of such wind farms will introduce many new challenges. In particular if the wind turbines can be moved slightly to reduce wake effect.

Acknowledgements

The work of the first author was carried out as an industrial PhD project, supported by Innovation Fund Denmark and Vattenfall BA Wind, in collaboration with DTU Management. We would like to thank our industrial partner, and in the specific Jesper Runge Kristoffersen, Iulian Vranceanu, Thomas Hjort and Kenneth Skaug who helped us in defining the cable routing and wind park design constraints; and Per Christian Hyldahl who helped defining jacket foundation problem and testing our model.

References

Archer, R., G. Nates, S. Donovan, and H. Waterer (2011). "Wind Turbine Interference in a Wind Farm Layout Optimization – Mixed Integer Linear Programming Model". In: *Wind Engineering* 35, no.2, pp. 165–178.

Barthelmie, R., K. Hansen, S. T. Frandsen, O. Rathmann, J. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. Politis, and P. K. Chaviaropoulos (2009). "Modelling and measuring flow and wind turbine wakes in large wind farms offshore". In: *Wind Energy* 12, pp. 431–444.

Bauer, J. and J. Lysgaard (2015). "The offshore wind farm array cable layout problem: a planar open vehicle routing problem". In: *Journal of the Operational Research Society* 66.3, pp. 360–368.

Berzan, C., K. Veeramachaneni, J. McDermott, and U. O. Reilly (2011). "Algorithms for cable network design on large-scale wind farms". In: *Tech. Rep. Tufts University*.

Dai, L., M. Stålhane, and I. B. Utne (2015). "Routing and Scheduling of Maintenance Fleet for Offshore Wind Farms". In: *Wind Engineering* 39.1, pp. 15–30.

DNV-GL (2005). "Fatigue design of offshore steel structures". In: *Recommended Practice DNVGL-RP-C203:2014-06*.

Dutta, S. and T. J. Overbye (2011). "A Clustering based Wind Farm Collector System Cable Layout Design". In: *Power and Energy Conference at Illinois (PECI)*, pp. 1–6.

EWEA (2012). *European Wind Energy Association*. `http://www.ewea.org/`.

Fagerfjall, P. (2010). "Optimizing Wind Farm Layout - More Bang for the Buck Using Mixed Integer Linear Programming". MA thesis. Goteborg, Sweden: Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University.

Fischetti, M. (2014). "Mixed-Integer Models and Algorithms for Wind Farm Layout Optimization". `http://tesi.cab.unipd.it/45458/1/tesi_Fischetti.pdf`. MA thesis. University of Padova.

Fischetti, M. and A. Lodi (2011). "Heuristics in Mixed Integer Programming". In: *Wiley Encyclopedia of Operations Research and Management Science (James J. Cochran ed.) Vol. 8*. John Wiley and Sons, pp. 738–747.

Fischetti, M. and M. Monaci (2014). "Proximity search for 0-1 mixed-integer convex programming". In: *Journal of Heuristics* 6.20, pp. 709–731.

Fischetti, M. and M. Monaci (2015). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics*. ISSN: 1381-1231.

Fischetti, M. and D. Pisinger (2017a). "Mixed Integer Linear Programming for new trends in wind farm cable routing". In: *ENDM Special Issue for INOC2017*.

Fischetti, M. and D. Pisinger (2017b). "Optimizing wind farm cable routing considering power losses". In: *European Journal of Operational Research*. ISSN: 0377-2217. DOI: http://dx.doi.org/10.1016/j.ejor.2017.07.061. URL: http://www.sciencedirect.com/science/article/pii/S037722171730704X.

Fischetti, M. and D. Pisinger (2017c). "On the Impact of using Mixed Integer Programming Techniques on Real-world Offshore Wind Parks." In: *ICORES*, pp. 108–118.

Fischetti, M. and D. Pisinger (2018). "Mathematical Optimization for offshore wind farm design: an overview". In: *Business and Information Systems Engineering*. DOI: 10.1007/s12599-018-0538-0.

Gundegjerde, C., I. B. Halvorsen, E. E. Halvorsen-Weare, L. M. Hvattum, and L. M. Nonås (2015). "A stochastic fleet size and mix model for maintenance operations at offshore wind farms". In: *Transportation Research Part C: Emerging Technologies* 52.Supplement C, pp. 74–92.

Gutierrez-Alcoba, A., G. Ortega, E. M. Hendrix, and E. E. (2017). "A model for optimal fleet composition of vessels for offshore wind farm maintenance". In: *Procedia Computer Science* 108.Supplement C. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland, pp. 1512–1521. ISSN: 1877-0509.

Hou, P., P. Enevoldsen, J. Eichman, W. Hu, M. Z. Jacobson, and Z. Chen (2017). "Optimizing investments in coupled offshore wind-electrolytic hydrogen storage systems in Denmark". In: *Journal of Power Sources* 359, pp. 186–197.

Hou, P., W. Hu, M. Soltani, B. Zhang, and Z. Chen (2016). "Optimization of decommission strategy for offshore wind farms". In: IEEE.

Irawan, C. A., D. Ouelhadj, D. Jones, M. Stålhane, and I. B. Sperstad (2017). "Optimisation of maintenance routing and scheduling for offshore wind farms". In: *European Journal of Operational Research* 256.1, pp. 76–89.

Jensen, N. (1983). *A note on wind generator interaction*. Tech. rep. Technical Report Riso-M-2411(EN), Riso National Laboratory, Roskilde, Denmark.

Kristoffersen, J. and P. Christiansen (2003). "Horns Rev offshore windfarm: Its main controller and remote control system". In: *Wind Engineering* 27.5, pp. 351–359.

Kusiak, A. and Z. Song (2010). "Design of wind farm layout for maximum wind energy capture". In: *Renewable Energy* 35.3, pp. 685–694.

Kwong, W., P.Y., R. D., and a. A. C. Moran J. and Michael M. (2012). "Multi-objective optimization of wind farm layouts under energy generation and noise propagation". In: *Proceedings of the ASME 2012 IDETC/CIE 2012, Chicago*.

Muskulus, M. and S. Schafhirt (2014). "Design optimization of wind turbine support structures-a review". In: *Journal of Ocean and Wind Energy* 1.1, pp. 12–22.

Oest, J., R. Sørensen, L. C. T. Overgaard, and E. Lund (2017). "Structural optimization with fatigue and ultimate limit constraints of jacket structures for large offshore wind turbines". In: *Structural and Multidisciplinary Optimization* 55.3, pp. 779–793.

Sinha, S. and S. Chandel (2015). "Review of recent trends in optimization techniques for solar photovoltaic–wind based hybrid energy systems". In: *Renewable and Sustainable Energy Reviews* 50.Supplement C, pp. 755–769.

Topham, E. and D. McMillan (2017). "Sustainable decommissioning of an offshore wind farm". In: *Renewable Energy* 102, pp. 470–480.

Turner, S., D. Romero, P. Zhang, C. Amon, and T. Chan (2014). "A new mathematical programming approach to optimize wind farm layouts". In: *Renewable Energy* 63.C, pp. 674–680.

Vattenfall (2017). *internal image storage.*

Zhang, P. Y., D. A. Romero, J. C. Beck, and C. H. Amon (2014). "Solving wind farm layout optimization with mixed integer programs and constraint programs". In: *EURO Journal on Computational Optimization* 2.3, pp. 195–219.

Chapter 3

Matheuristics

Martina Fischetti[a] · Matteo Fischetti[b] · Michele Monaci[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]DEI, Dipartimento di Ingegneria dell'Informazione University of Padova, via Gradenigo 6/A, 35131 Padova, Italy

[c]Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione Guglielmo Marconi, Viale del Risorgimento 2, 40136 Bologna, Italy

Publication Status: Based on the book chapters: *Fischetti and Fischetti, 2016* and *Fischetti et al., 2016*

Reading Instructions: Detailed description of the MILP-based heuristic techniques we used in our work. To be used as background information.

Abstract: As its name suggests, a matheuristic is the hybridization of mathematical programming with metaheuristics. The hallmark of matheuristics is the central role played by the mathematical programming model, around which the overall heuristic is built. As such, matheuristic is not a rigid paradigm, but rather a concept framework for the design of mathematically-sound heuristics. The aim of this chapter is to introduce the main matheuristic ideas. Three specific applications in the field of wind farm, packing, and vehicle routing optimization, respectively, are addressed and used to illustrate the main features of the method.

3.1 Introduction

The design of heuristics for difficult optimization problems is itself a heuristic process that often involves the following main steps.

After a clever analysis of the problem at hand and of the acceptable simplifications in its definition, one tries to set up an effective **Mathematical Programming** (MP) model and to solve it by a general purpose piece of software—often a Mixed-Integer Linear Programming (MIP) solver. Due to the impressive improvement of general-purpose solvers in recent years, this approach can actually solve the instances of interest to proven optimality (or with an acceptable approximation) within a reasonable computing time, in which case of course no further effort is needed.

If this is not the case, one can insist on the MP approach and try to obtain better and better results by improving the model and/or by enhancing the solver by specialized features (cutting planes, branching, etc.). Or one can forget about MP, and resort to ad-hoc heuristics not based on the MP model. In this latter case, the MP model is completely disregarded, or just used for illustrating the problem characteristics and/or for getting an off-line indication of the typical approximation error on a set of sample instances.

A third approach is however possible, that consists in using the MP solver as a basic tool *within* the heuristic framework. This hybridization of MP with **Metaheuristics** leads to the ***Matheuristic*** approach, where the heuristic is built around the MP model. Matheuristics became popular in recent years, as witnessed by the publication of dedicated volumes and journal special issues Hansen et al., 2009; Fischetti and Lodi, 2011; Maniezzo et al., 2010 and by the dedicated sessions on MP and metaheuristic conferences.

Designing an effective heuristic is an art that cannot be framed into strict rules. This is particularly true when addressing a matheuristic, which is not a rigid paradigm but a concept framework for the design of mathematically-sound heuristics. In the present chapter we will therefore try to illustrate some main matheuristic features with the help of different examples of application.

The present chapter is based on previous published work (i.e. mainly, Fischetti and Lodi, 2011, Fischetti and Monaci, 2016) .

3.2 General-purpose MIP-based heuristics

Heuristics for general-purpose MIP solvers form the basis of the matheuristic's toolkit. Their relevance for our chapter is twofold. On the one hand, they are invaluable tools for the solution of the subproblems tailored by the matheuristic when applied to a specific problem. On the other hand, they illustrate the benefits for a general-purpose MIP solver deriving from the use of metaheuristics concepts such as local search and evolutionary methods.

Modern MIP solvers exploit a rich arsenal of tools to attack hard problems. It is widely accepted that the solution of hard MIPs can take advantage from the solution of a series of auxiliary Linear Programs (LPs) intended to enhance the performance of the overall MIP solver. E.g., auxiliary LPs may be solved to generate powerful disjunctive cuts, or to implement a strong branching policy. On the other hand, it is a common experience that finding good-quality heuristic MIP solutions often requires a computing time that is just comparable to that needed to solve the LP relaxation. So, it makes sense to think of exact/heuristic MIP solvers where auxiliary MIPs (as opposed to LPs) are heuristically solved on the fly, with the aim of bringing the MIP technology under the chest of the MIP solver itself. This leads to the idea of "translating into a MIP model" (*MIPping* in the jargon of Fischetti et al., 2010) some crucial decisions to be taken when designing a MIP-based algorithm.

We next describe the new generation of MIP heuristics that emerged in the late 1990s, which are based on the idea of systematically using a "black-box" external MIP solver to explore a solution neighborhood defined by invalid linear constraints. We address a generic MIP of the form

$$(MIP) \qquad \min c^T x \tag{3.1}$$

$$Ax \geq b, \tag{3.2}$$

$$x_j \in \{0,1\},\ \forall j \in \mathcal{B}, \tag{3.3}$$

$$x_j \text{ integer}, \ \forall j \in \mathcal{G}, \tag{3.4}$$
$$x_j \text{ continuous}, \ \forall j \in \mathcal{C}, \tag{3.5}$$

where A is an $m \times n$ input matrix, and b and c are input vectors of dimension m and n, respectively. Here, the variable index set $\mathcal{N} := \{1, \ldots, n\}$ is partitioned into $(\mathcal{B}, \mathcal{G}, \mathcal{C})$, where $\mathcal{B}$ is the index set of the 0-1 variables (if any), while sets $\mathcal{G}$ and $\mathcal{C}$ index the general integer and the continuous variables, respectively. Removing the integrality requirement on variables indexed by $\mathcal{I} := \mathcal{B} \cup \mathcal{G}$ leads to the so-called *LP relaxation*.

3.2.1 Local Branching

The ***Local Branching*** (LB) scheme of Fischetti and Lodi Fischetti and Lodi, 2003 appears to be one of the first general-purpose heuristics using a black-box MIP solver applied to subMIPs, and it can be viewed as a precursor of matheuristics. Given a *reference solution* $\bar{x}$ of a MIP with $\mathcal{B} \neq \emptyset$, one aims at finding an improved solution that is "not too far" from $\bar{x}$, in the sense that not too many binary variables need be flipped. To this end, one can define the k-opt neighborhood $\mathcal{N}(\bar{x}, k)$ of $\bar{x}$ as the set of the MIP solutions satisfying the invalid *local branching constraint*

$$\Delta(x, \bar{x}) := \sum_{j \in \mathcal{B}: \bar{x}_j = 0} x_j + \sum_{j \in \mathcal{B}: \bar{x}_j = 1} (1 - x_j) \leq k, \tag{3.6}$$

for a small neighborhood radius k—an integer parameter typically set to 10 or 20. The neighborhood is then explored (possibly heuristically, i.e., with some small node or time limit) by means of a black-box MIP solver. Experimental results Fischetti and Monaci, 2014 show that the introduction of the local branching constraint typically has the positive effect of driving to integrality many component of the optimal solution of the LP relaxation, improving the so-called "relaxation grip" and hence the capability of the MIP solver to find (almost) optimal integer solutions within short computing times. Of course, this effect is lost if parameter k is set to a large value—a mistake that would make local branching completely ineffective.

LB is in the spirit of local search metaheuristics and in particular of *Large Neighborhood Search* (LNS) Shaw, 1998, with the novelty that neighborhoods are obtained through "soft fixing", i.e., through invalid cuts to be added to the original MIP model. Diversification cuts can be defined in a similar way, thus leading to a flexible toolkit for the definition of metaheuristics for general MIPs.

3.2.2 Relaxation Induced Neighborhood Search

The ***Relaxation Induced Neighborhood Search*** (RINS) heuristic of Danna, Rothberg and Le Pape Danna et al., 2005a also uses a black-box MIP solver to explore a neighborhood of a given solution $\bar{x}$, and was originally designed to be integrated in a branch-and-bound solution scheme. At specified nodes of the branch-and-bound tree, the current LP relaxation solution x^* and the incumbent $\bar{x}$ are compared and all integer-constrained variables that agree in value are fixed. The resulting MIP is typically easy to solve, as fixing reduces its size considerably, and often provides improved solutions with respect to $\bar{x}$.

3.2.3 Polishing a feasible solution

The ***Polishing*** algorithm of Rothberg Rothberg, 2007 implements an evolutionary MIP heuristic which is invoked at selected nodes of a branch-and-bound tree and includes all classical ingredients of genetic computation, namely:

- *Population*: A fixed-size population of feasible solutions is maintained. Those solutions are either obtained within the branch-and-bound tree (by other heuristics), or computed by the polishing algorithm itself.

- *Combination*: Two or more solutions (the parents) are combined with the aim of creating a new member of the population (the child) with improved characteristics. The RINS scheme is adopted, i.e., all variables whose value coincides in the parents are fixed and the reduced MIP is heuristically solved by a black-box MIP solver within a limited number of branch-and-bound nodes. This scheme is clearly much more time-consuming than a classical combination step in evolutionary algorithms, but it guarantees feasibility of the child solution.

- *Mutation*: Diversification is obtained by performing a classical mutation step that (i) randomly selects a "seed" solution in the population, (ii) randomly fixes some of its variables, and (iii) heuristically solves the resulting reduced MIP.

- *Selection*: Selection of the two parents to be combined is performed by randomly picking a solution in the population and then choosing, again at random, the second parent among those solutions with a better objective value.

3.3 Proximity search heuristics for general mixed-integer programs

Proximity search is a general approach aimed at improving a given feasible "reference solution", quickly producing a sequence of solutions of improved quality. For the sake of being self-contained, we next outline the main features of this technique, and refer the reader to Fischetti and Monaci, 2014 for fuller details.

Proximity search is related to *Large-Neighborhood Search* (LNS) heuristics Shaw, 1998, that also explore a solution neighborhood defined by invalid constraints. For instance, *Local Branching* Fischetti and Lodi, 2003 adds a constraint that limits the search to solutions that are "sufficiently close" to the reference solution. Similarly, *Relaxation Induced Neighborhood Search* (RINS) Danna et al., 2005b is a heuristic that addresses the neighborhood resulting from fixing all variables having the same value in the reference and in relaxation solutions. Proximity search is also related to the parametric branch-and-bound algorithm proposed by Glover Glover, 1978 and extended in Glover, 2006 to a parametric tabu search algorithm. A different approach that defines a neighborhood of a given solution is the *Feasibility Pump* paradigm introduced in Fischetti et al., 2005 for 0-1 Mixed-Integer Linear Programs, and extended to nonlinear problems in Bonami et al., 2009 and D'Ambrosio et al., 2012 (among others).

A distinguished feature of proximity search is that no invalid constraint is added to the model, but the objective function is modified to favor solutions that are "close enough" to the reference solution. The idea proved quite effective in quickly improving a given starting feasible solution, at least when the landscape of feasible solutions is not too irregular—as it happens in the wind farm layout optimization context.

3.3.1 The basic idea

For the sake of generality, in this section we will focus on a generic 0-1 Mixed-Integer (possibly nonlinear) Program of the form

$$\min f(x) \tag{3.7}$$

$$g(x) \leq 0 \tag{3.8}$$

$$x_j \in \{0,1\} \quad \forall j \in J \tag{3.9}$$

Algorithm 1 The basic Proximity Search heuristic

1: let $\tilde{x}$ be the initial heuristic feasible solution to refine;
2: **while** an overall termination condition is not reached **do**
3: explicitly add the *cutoff constraint* $f(x) \leq f(\tilde{x}) - \theta$ to the MIP model;
4: replace $f(x)$ by the proximity objective function $\Delta(x, \tilde{x})$;
5: run the MIP solver on the new model until a termination condition is reached, and let x^* be the best feasible solution found (x^* empty if none);
6: **if** x^* is nonempty and $J \subset N$ **then**
7: refine x^* by solving the convex program $x^* := \text{argmin}\{f(x) : g(x) \leq 0,\ x_j = x_j^*\ \forall j \in J\}$
8: **end if**
9: recenter $\Delta(x, \cdot)$ by setting $\tilde{x} := x^*$, and/or update θ
10: **end while**
11: **return** $\tilde{x}$

where $f : \Re^n \to \Re$, $g : \Re^n \to \Re^m$, and $J \subseteq N := \{1, \ldots, n\}$, $J \neq \emptyset$, indexes binary variables. Although this is not strictly required by the method, in the following we assume that both f and g are convex functions with the property that dropping the integrality condition in (3.9) leads to a polynomially solvable relaxation.

Proximity search starts with a feasible solution $\tilde{x}$, and modifies the MIP formulation as follows:

- add an explicit *cutoff constraint*

$$f(x) \leq f(\tilde{x}) - \theta, \tag{3.10}$$

  where $\theta > 0$ is a given cutoff tolerance; and

- replace the original objective function with the Hamming distance

$$\Delta(x, \tilde{x}) := \sum_{j \in J:\, \tilde{x}_j = 0} x_j + \sum_{j \in J:\, \tilde{x}_j = 1} (1 - x_j) \tag{3.11}$$

The conceptual scheme of proximity search is sketched in Algorithm 1.

Proximity search requires an initial solution $\tilde{x}$ at Step 1. This solution can be computed by using some ad-hoc heuristic or by running the black-box MIP solver from scratch until a first feasible solution is found. In any case, we

assume that finding a feasible solution is not really an issue for the problem at hand. If this is not the case, one should resort to a problem reformulation where some constraints are imposed in a soft way through violation penalties attached to slack variables.

At Step 3, the cutoff tolerance θ is defined and a the cutoff constraint is added.

At Step 4, the objective function is redefined as the Hamming distance between x and $\tilde{x}$, see (3.11). Then, the resulting problem is solved by using a black-box MIP solver to hopefully find a new feasible solution, say x^*. If this is the case, it must be $f(x^*) \leq f(\tilde{x}) - \theta$, i.e., x^* improves over $\tilde{x}$.

A key property of the approach is that the root-node solution of the convex relaxation, say x', is expected to be not too different from $\tilde{x}$, as this latter solution would be optimal without the cutoff constraint. As a matter of fact, for a small θ this constraint can typically be fulfilled with just minor adjustments of $\tilde{x}$, a property that is instrumental for the success of the method because of two main positive effects:

- the computing time spent at the root node is often very small (even orders of magnitude smaller than the time required at the root node for the original problem);
- solution x' is typically "almost integer", i.e., with a small number of fractional components indexed by J, thus improving the chances of success of the MIP internal heuristics.

Table 3.1 is taken from Fischetti and Monaci, 2014, and illustrates both positive effects for the set covering (pure binary) MIPLIB2010 instance `ramos3` when considering a reference solution $\tilde{x}$ of value 267. The table reports, for different values of parameter θ, the number of components of the LP relaxation solution x' that belong to the intervals [0,0], (0, 0.1], ... (0.9, 1], and [1,1], along with computing time (in CPU sec.s), number of simplex iterations (dual pivots), and objective value—i.e., distance $\Delta(x', \tilde{x})$. The LP relaxation becomes infeasible for $\theta > 121$. The table shows that, for small values of θ, the LP-solution time is just negligible, while the number of integer components is very large. On the contrary, using a too large value for θ leads to LP-solutions that are "far away" from $\tilde{x}$ in terms of distance, have a large number of fractional components, and require a considerable computational effort to be computed—thus vanishing all proximity search benefits.

If no new solution x^* is found at Step 5 (possibly because the MIP solver was

Table 3.1: Distribution of fractionalities in the LP relaxation solution x' and corresponding computing time and distance $\Delta(x', \tilde{x})$ from the reference solution $\tilde{x}$, for various values of the cutoff parameter θ.

x-range	$\theta = 0$	$\theta = 1$	$\theta = 2$	$\theta = 3$	$\theta = 4$	$\theta = 5$	$\theta = 10$	$\theta = 20$	$\theta = 30$	$\theta = 50$	$\theta = 99$	$\theta = 121$
$= 0$	1920	1919	1919	1919	1924	1920	1619	1619	1600	1565	1276	682
$(0.0, 0.1]$	0	0	0	0	0	0	303	297	293	281	420	926
$(0.1, 0.2]$	0	0	0	0	0	4	0	6	26	65	194	380
$(0.2, 0.3]$	0	1	0	5	0	0	0	3	7	15	64	169
$(0.3, 0.4]$	0	0	0	0	0	0	0	1	2	8	75	29
$(0.4, 0.5]$	0	0	6	0	0	0	8	4	3	16	91	0
$(0.5, 0.6]$	0	0	0	0	0	0	5	5	9	19	47	1
$(0.6, 0.7]$	0	0	0	0	0	0	0	2	9	35	17	0
$(0.7, 0.8]$	0	5	0	1	0	1	0	10	25	88	3	0
$(0.8, 0.9]$	0	0	0	0	0	11	0	28	101	68	0	0
$(0.9, 1.0)$	0	0	0	0	0	0	249	209	110	26	0	0
$= 1$	267	262	262	262	263	251	3	3	2	1	0	0
time (sec.s)	0.00	0.04	0.03	0.03	0.04	0.21	0.45	0.54	0.57	0.90	4.77	30.91
# LP-iter.s	0	352	341	357	358	1180	2164	2543	2637	3627	6829	11508
Δ-distance	0.00	1.50	3.00	4.50	6.00	7.88	17.45	37.13	56.86	96.90	208.71	292.67

aborted before convergence), one proceeds directly to Step 9 where tolerance θ is reduced. Of course, if the MIP solver proved infeasibility for the given θ, one has that $f(\tilde{x}) - \theta$ is a valid lower bound on the optimal value of the original MIP.

Step 7 is aimed at improving the new solution x^*, if any, by solving a convex problem where all binary variables have been fixed to their value in x^*; in this way, the best solution within the neighborhood induced by $\Delta(x, x^*) = 0$ is determined.

At Step 9, the approach is iterated using the current incumbent $\tilde{x}$ (if available) so as to recenter the distance function Δ, and/or by modifying the cutoff tolerance θ.

3.3.2 Proximity search implementations

In this section we sketch three possible implementations of the basic proximity search method, as described in Fischetti and Monaci, 2014. All three implementations start with a given solution $\tilde{x}$, replace the original objective function $f(x)$ by a proximity one $\Delta(x, \cdot)$, and use a cutoff constraint to force the detection of improved solutions.

In the first implementation, denoted as "without recentering" and described in Subsection 3.3.2.1, the proximity objective function $\Delta(x, \cdot)$ remains centered on the very first solution $\tilde{x}$, while the cutoff constraint is modified on the fly inside the MIP solver.

The second implementation, called "with recentering" (see Subsection 3.3.2.2), is an iterative scheme in which the MIP solver is halted as soon as an improved solution, say x', is found. In this case, a new problem is defined by replacing $\tilde{x}$ by x' both in the objective $\Delta(x, \cdot)$ and in the cutoff constraint, and the MIP solver is re-applied from scratch.

A variant of the second implementation is given in Subsection 3.3.2.3. We call it "with incumbent", as the cutoff constraint is imposed in a soft way to make the initial solution $\tilde{x}$ feasible—though highly penalized in the objective function. In this way the current incumbent $\tilde{x}$ can be used for a warm-start of MIP solver.

3.3.2.1 Proximity search without recentering

This version of proximity search assumes that the MIP solver can be controlled through a callback function to be executed each time the incumbent is going to be updated—this is a standard feature of modern MIP solvers. Within such a function, the new incumbent $\hat{x}$ (say) is first internally recorded in a user's data structure, and a new global cut

$$f(x) \leq f(\hat{x}) - \theta \tag{3.12}$$

is added to the current model. The new cutoff (3.12) makes solution $\hat{x}$ infeasible, thus preventing the solver to update its own incumbent, and forces search to explore only solutions improving over $\hat{x}$.

Inequality (3.12) is imposed for the initial solution $\tilde{x}$ as well, making it infeasible. In this way the optimal relaxation solution x' at the root node is different from $\tilde{x}$, and violated MIP cuts can possibly be generated at the root node.

The simple implementation above has a main advantage in that a single enumeration tree is generated. However, the are some main drawbacks that can affect the performance of the method in a negative way, namely:

- The proximity objective function is always computed with respect to the initial solution $\tilde{x}$ and is not changed during the search. Thus, $\Delta(x, \cdot)$ remains "centered" with the very first solution (hence the name "proximity search without recentering"), though after some enumeration one has to explore a solution space that is far away from $\tilde{x}$.

- The MIP incumbent is never explicitly updated, which prevents the application of powerful propagation and variable-fixing schemes—as well as refinement heuristics—embedded in the MIP solver.
- The scheme requires to install some callback functions, using the MIP solver as a grey-box, which may deactivate some of its features.

3.3.2.2 Proximity search with recentering

As already noticed, the scheme presented in the previous section has some drawbacks, mainly related to the need of interacting with the underlying MIP and with the choice to keep the objective function centered with the first solution $\tilde{x}$. We now present a different implementation that uses the MIP solver as a black box (with no need of callback functions) and just restarts it as soon as a new solution is found.

In the new implementation, called "proximity search with recentering", Steps 1 to 4 are the same as in Algorithm 1. After the new problem has been defined, one invokes the MIP solver as a black box (Step 5), in its default mode and without any callback, and aborts its execution as soon as a first feasible solution is found. Due to the cutoff constraint in the model, this solution (if any) is a strict improvement over $\tilde{x}$. At Step 9 the method is then iterated, by replacing $\tilde{x}$ with the new solution and repeating (without changing θ) from Step 3, until the overall time limit is reached. Obviously, in case no solution is found at Step 5, the algorithm either proves θ-optimality of the incumbent $\tilde{x}$ or hits the given time limit.

We observe that this scheme has the main advantage to use the MIP solver as a black-box (in its default settings, without callbacks), and that it implements a dynamic updating of the objective function. In addition, the scheme can be implemented very easily. A main disadvantage of the method is that overlapping search trees are explored, possibly wasting computing time and solving the (computationally heavy) root node of the problem many times.

3.3.2.3 Proximity search with incumbent

The third implementation we address is aimed at solving a drawback that is common for both implementations above; namely, adding the cutoff constraints prevents the MIP solver to update its internal incumbent and to apply some

powerful refinement heuristics such as RINS Danna et al., 2005b. In the "proximity search with recentering" variant one imposed the cutoff constraint (3.10) in its "soft version"

$$f(x) \leq f(\tilde{x}) - \theta + z \tag{3.13}$$

where $z \geq 0$ is a continuous slack variable. Variable z is highly penalized in the objective function, which is modified to

$$\Delta(x, \tilde{x}) + Mz \tag{3.14}$$

where M is a large positive value compared to the feasible values of Δ. With this formulation, solution $\tilde{x}$ can be provided on input to the MIP solver as a feasible warm-start solution with $z = 1$. Though the cost of this solution is very large, having an internal incumbent allows the MIP solver to trigger its internal refinement heuristics. In this case too, whenever a new incumbent is found with $z = 0$, i.e., a θ-improving solution has been found, the execution is aborted and the method is iterated.

References

Bonami, P., G. Cornuéjols, A. Lodi, and F. Margot (2009). "A Feasibility Pump for mixed integer nonlinear programs". In: *Mathematical Programming* 119.2, pp. 331–352. ISSN: 1436-4646. DOI: 10.1007/s10107-008-0212-2. URL: https://doi.org/10.1007/s10107-008-0212-2.

D'Ambrosio, C., A. Frangioni, L. Liberti, and A. Lodi (2012). "A storm of feasibility pumps for nonconvex MINLP". In: *Mathematical Programming* 136.2, pp. 375–402. ISSN: 1436-4646. DOI: 10.1007/s10107-012-0608-x. URL: https://doi.org/10.1007/s10107-012-0608-x.

Danna, E., E. Rothberg, and C. L. Pape (2005a). "Exploring relaxation induced neighborhoods to improve MIP solutions". In: *Mathematical Programming*, pp. 71–90.

Danna, E., E. Rothberg, and C. L. Pape (2005b). "Exploring relaxation induced neighborhoods to improve MIP solutions". In: *Mathematical Programming* 102.1, pp. 71–90. DOI: 10.1007/s10107-004-0518-7. URL: https://doi.org/10.1007/s10107-004-0518-7.

Fischetti, M. and A. Lodi (2003). "Local Branching". In: *Mathematical Programming*, pp. 23–47.

Fischetti, M. and A. Lodi (2011). "Heuristics in Mixed Integer Programming". In: *Wiley Encyclopedia of Operations Research and Management Science (James J. Cochran ed.) Vol. 8*. John Wiley and Sons, pp. 738–747.

Fischetti, M., A. Lodi, and D. Salvagnin (2010). "Just MIP it!" English. In: *Matheuristics*. Ed. by V. Maniezzo, T. Stützle, and S. Voß. Vol. 10. Annals of Information Systems. Springer US, pp. 39–70. ISBN: 978-1-4419-1305-0. DOI: 10.1007/978-1-4419-1306-7_2. URL: http://dx.doi.org/10.1007/978-1-4419-1306-7_2.

Fischetti, M. and M. Monaci (2014). "Proximity search for 0-1 mixed-integer convex programming". In: *Journal of Heuristics* 6.20, pp. 709–731.

Fischetti, M. and M. Fischetti (2016). "Matheuristics". In: *Handbook of Heuristics*. Ed. by R. Martí, P. Panos, and M. G. Resende. Springer International Publishing, pp. 1–33. ISBN: 978-3-319-07153-4. DOI: 10.1007/978-3-319-07153-4_14-1. URL: http://dx.doi.org/10.1007/978-3-319-07153-4_14-1.

Fischetti, M., M. Fischetti, and M. Monaci (2016). "Optimal Turbine Allocation for Offshore and Onshore Wind Farms". In: *Optimization in the Real World: Toward Solving Real-World Optimization Problems*. Ed. by K. Fujisawa, Y. Shinano, and H. Waki. Tokyo: Springer Japan, pp. 55–78. ISBN: 978-4-431-55420-2. DOI: 10.1007/978-4-431-55420-2_4. URL: https://doi.org/10.1007/978-4-431-55420-2_4.

Fischetti, M. and M. Monaci (2016). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: 10.1007/s10732-015-9283-4. URL: https://doi.org/10.1007/s10732-015-9283-4.

Fischetti, M., F. Glover, and A. Lodi (2005). "The feasibility pump". In: *Mathematical Programming* 104.1, pp. 91–104. ISSN: 1436-4646. DOI: 10.1007/s10107-004-0570-3. URL: https://doi.org/10.1007/s10107-004-0570-3.

Glover, F. (1978). "Parametric branch and bound". In: *Omega* 6.2, pp. 145–152. ISSN: 0305-0483. DOI: https://doi.org/10.1016/0305-0483(78)90022-1. URL: http://www.sciencedirect.com/science/article/pii/0305048378900221.

Glover, F. (2006). "Parametric tabu-search for mixed integer programs". In: *Computers & Operations Research* 33.9. Part Special Issue: Anniversary Focused Issue of Computers & Operations Research on Tabu Search, pp. 2449–2494. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2005.07.009. URL: http://www.sciencedirect.com/science/article/pii/S0305054805002273.

Hansen, P., V. Maniezzo, and S. Voß (2009). "Special issue on mathematical contributions to metaheuristics editorial". In: *Journal of Heuristics* 15.3, pp. 197–199. URL: http://dx.doi.org/10.1007/s10732-008-9093-z.

Maniezzo, V., T. Stützle, and S. Voß, eds. (2010). *Matheuristics - Hybridizing Metaheuristics and Mathematical Programming*. Vol. 10. Annals of Informa-

tion Systems. Springer. ISBN: 978-1-4419-1305-0. URL: http://dx.doi.org/10.1007/978-1-4419-1306-7.

Rothberg, E. (2007). "An Evolutionary Algorithm for Polishing Mixed Integer Programming Solutions". In: *INFORMS Journal on Computing*, pp. 534–541.

Shaw, P. (1998). "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems". In: *Principles and Practice of Constraint Programming — CP98*. Ed. by M. Maher and J.-F. Puget. Vol. 1520. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 417–431. ISBN: 978-3-540-65224-3. DOI: 10.1007/3-540-49481-2_30. URL: http://dx.doi.org/10.1007/3-540-49481-2_30.

Part II

Wind Farm Layout

Chapter 4

Proximity search heuristics for wind farm optimal layout

Martina Fischetti[a] · Michele Monaci[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione Guglielmo Marconi, Viale del Risorgimento 2, 40136 Bologna, Italy

Publication Status: Published as *Fischetti and Monaci, 2016*

Reading Instructions: Main paper on the wind farm layout optimization problem.

Abstract: A heuristic framework for turbine layout optimization in a wind farm is proposed that combines ad-hoc heuristics and Mixed-Integer

Linear Programming. In our framework, large-scale Mixed-Integer Programming models are used to iteratively refine the current best solution according to the recently-proposed *proximity search* paradigm. Computational results on very large scale instances involving up to 20,000 potential turbine sites prove the practical viability of the overall approach.

4.1 Introduction

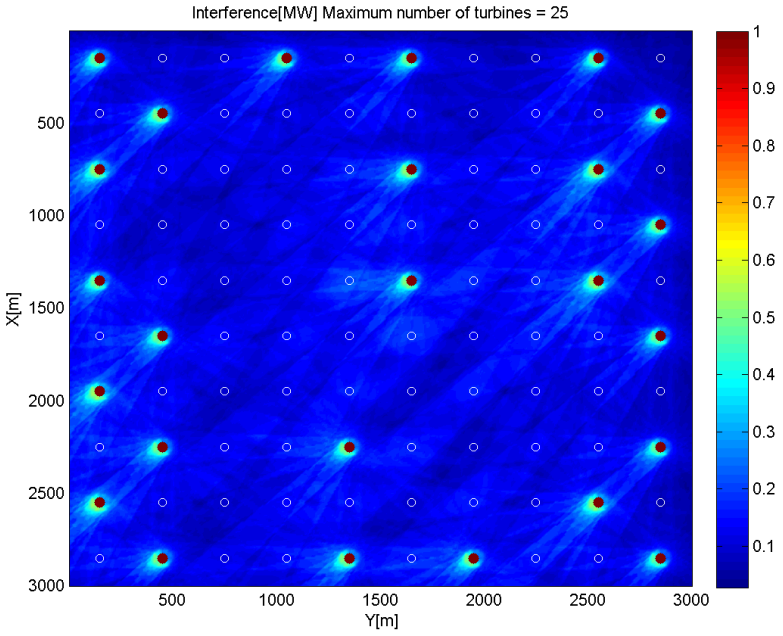
Green energy became a topic of great interest in recent years. Indeed, environmental sustainability asks for a considerable reduction in the use of fossil fuels, that are pollutant and unsustainable. As a consequence, ambitious plans have been proposed for green energy production, including wind energy. The *wind farm layout optimization problem* consists in finding an optimal allocation of turbines in a given site, to maximize the power output. This strategic problem is extremely hard in practice, both for the sizes of the instances in real applications and for the presence of several nonlinearities to be taken into account. A typical nonlinear feature of this problem is the interaction among turbines, also known as *wake effect*. The wake effect is the interference phenomenon for which, if two turbines are located one close to another, the upwind one creates a shadow on the one behind. This is of great importance in the design of the layout since it results in a loss of power production for the turbine downstream, that is also subject to a possibly strong turbulence.

It is estimated in Barthelmie et al., 2009 that in large offshore wind farms, the average power loss due to turbine wakes is around 10-20% of the total energy production. It is then obvious that power production can increase significantly if the farm layout is designed so as to reduce the effect of turbine wake as much as possible.

Figure 4.1 illustrates the wind farm problem corresponding to a 3,000 $\times$ 3,000 (m) offshore area where turbines can be installed. The small circles identify the points where a turbine can potentially be built (*sites*), while filled circles refer to the currently built turbines. Interference due to the built turbines are represented in the background of the figure, and refer to the average interference over 500 macro-scenarios computed on real-world wind data from Vattenfall AB Vattenfall, 2014.

As mentioned, interference plays a relevant role in the definition of the problem. Different models have been proposed in the literature to define interference, the most common being kinematic and field models. Those in the former class only

Figure 4.1: Turbine packing in an offshore setting with cumulative interference (100 potential turbine locations on a regular 10×10 *grid).*



consider the velocity deficit of the wake behind a turbine, whereas field models compute the complete flow field through a wind farm. An exhaustive comparison between different models of interference is given in Renkema, 2007. In the present paper, we consider only the model proposed by Jensen Jensen, 1983 for computing the pairwise interference between a pair of turbines. In addition, we assume the overall interference be the sum of pairwise interferences; though the model is an approximation of the real context, it turns out to be accurate enough for our purposes. Indeed, this model is also used, e.g., in WindPRO, an industry-standard software for wind resource assessment and placement of wind turbines within wind farms EMD, n.d. Finally, a main advantage of this model is the possibility to implicitly deal with a large number wind scenarios, which is a must in practical cases. On the contrary, most of the alternative models for interference turn out to be impractical in these settings, as they require the definition of a large number of additional variables and constraints.

Our aim is to heuristically solve wind farm instances of large size, as arise in practical applications. To give the planners a reactive tool for their what-if

analyses, almost-optimal solutions should be computed in a matter of minutes on a standard PC—one hour being our time limit even for the largest cases. With this ambitious goal in mind, we investigated a novel approach that combines fast ad-hoc heuristics with a proximity-search Fischetti and Monaci, 2014 refinement procedure based on a compact Mixed-Integer Linear Programming (MIP) model. Computational results on a large benchmark of realistic instances are presented.

The paper is organized as follows. In Sect. 4.2 we introduce the MIP model used in our computation, which is designed as a compromise between Linear Programming (LP) relaxation tightness and compactness. A very fast ad-hoc heuristic is presented in Sect. 4.3, while the proximity search framework is outlined in Sect. 4.4. The overall scheme that combines our ad-hoc and MIP-based heuristics is described in Sect. 4.5, and computationally evaluated in Sect. 4.6. Finally, conclusions and directions of future research are addressed in Sect. 4.7.

The present paper is based on the first author's master thesis Fischetti, 2014.

4.2 Which MIP model?

We consider the problem in which the given offshore area has been sampled, and a number of possible positions for a turbine (called "sites" in what follows) has been identified. Alternative models in which a continuous layout is considered have been proposed in the literature (see, e.g., Kusiak and Song, 2010). However these models are highly nonconvex and turn out to be extremely challenging from a computational viewpoint. Thus, we considered a basic MIP model from the literature, which focuses on turbine distance constraints and on the wake effect (see, e.g., Donovan, 2005), and addresses the following constraints:

a) a minimum and maximum number of turbines that can be built is given;

b) there should be a minimal separation distance of between two turbines to ensure that the blades do not physically clash (turbine distance constraints);

c) if two turbines are installed, their interference will cause a loss in the power production that depends on their relative position and on wind conditions.

Let V denote the set of possible positions for a turbine, called "sites" in what follows, and let

- I_{ij} be the interference (loss of power) experienced by site j when a turbine is installed at site i, with $I_{jj} = 0$ for all $j \in V$;
- P_i be the power that a turbine would produce if built (alone) at site i;
- N_{MIN} and N_{MAX} be the minimum and maximum number of turbines that can be built, respectively;
- D_{MIN} be the minimum distance between two turbines;
- $dist(i, j)$ be the symmetric distance between sites i and j.

In addition, let $G_I = (V, E_I)$ denote the incompatibility graph with

$$E_I = \{[i, j] : i, j \in V,\, dist(i, j) < D_{MIN},\, j > i\}$$

and let $n := |V|$ denote the total number of sites.

Note that the interference matrix I is not symmetric, as the loss of power due to interference experienced by i when a turbine is installed in site j depends on the relative position of i with respect to j but also on the position of i with respect to the wind direction. In the model, two sets of binary variables are defined:

$$x_i = \begin{cases} 1 & \text{if a turbine is built at site } i \in V; \\ 0 & \text{otherwise} \end{cases} \qquad (i \in V)$$

$$z_{ij} = \begin{cases} 1 & \text{if two turbines are built at both sites } i \in V \text{ and } j \in V; \\ 0 & \text{otherwise} \end{cases} \qquad (i, j \in V, i < j)$$

The model then reads

$$\max \sum_{i \in V} P_i x_i - \sum_{i \in V} \sum_{j \in V, i<j} (I_{ij} + I_{ji}) z_{ij} \tag{4.1}$$

$$\text{s.t.} \quad N_{MIN} \leq \sum_{i \in V} x_i \leq N_{MAX} \tag{4.2}$$

$$x_i + x_j \leq 1 \qquad \forall [i, j] \in E_I \tag{4.3}$$

$$x_i + x_j - 1 \leq z_{ij} \qquad \forall i, j \in V, i < j \tag{4.4}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{4.5}$$

$$z_{ij} \in \{0,1\} \qquad \forall i,j \in V, i<j \tag{4.6}$$

Objective function (4.1) maximizes the total power production by taking interference losses I_{ij} into account. Constraints (4.4) force $z_{ij} = 1$ whenever $x_i = x_j = 1$; because of the objective function, this is in fact equivalent to setting $z_{ij} = x_i x_j$. Constraints (4.3) model pairwise site incompatibility, and can be strengthened to their clique counterpart

$$\textstyle\sum_{h\in Q} x_h \leq 1 \qquad \forall Q \in \mathcal{Q} \tag{4.7}$$

where $\mathcal{Q}$ is a family of maximal cliques of G_I, such that every edge in E_I is contained in at least one member of $\mathcal{Q}$. Constraints (4.6) can be relaxed to $z_{ij} \geq 0$, as integrality of the x variables implies the same property for the z.

The definition of the turbine power vector (P_i) and of interference matrix (I_{ij}) depends on the wind scenario considered, which greatly varies in time. Using statistical data, one can in fact collect a large number K of wind scenarios k, each associated with a pair (P^k, I^k) and with a probability π_k. Using that data, one can write a straightforward Stochastic Programming variant of the previous model where only the objective function needs to be modified into

$$\sum_{k=1}^{K} \pi_k \Big(\sum_{i\in V} P_i^k x_i - \sum_{i\in V} \sum_{j\in V, i<j} (I_{ij}^k + I_{ji}^k) z_{ij} \Big) \tag{4.8}$$

while all constraints stay unchanged as they only involve "first-stage" variables x and z. It is therefore sufficient to define

$$P_i := \sum_{k=1}^{K} \pi_k P_i^k \qquad \forall i \in V \tag{4.9}$$

$$I_{ij} := \sum_{k=1}^{K} \pi_k I_{ij}^k \qquad \forall i,j \in V \tag{4.10}$$

to obtain the same model (4.1)–(4.6) as before.

As already mentioned, assuming cumulative interference provides an approximated model, whose accuracy is however quite accurate (see, again, Jensen, 1983). Though more complex models of interference are available in the literature (see, e.g., Archer et al., 2011 and Renkema, 2007), we decided to stick to the model above for two main reasons: (i) the model is quite standard and well understood by practitioners Donovan, 2005, (ii) as we have just seen, a suitable definition of the input data allows one to easily address the realistic

situation in which many wind scenarios are considered; this is not the case for more sophisticated models, which typically lead to really huge stochastic programming variants.

While (4.1)–(4.6) turns out to be a reasonable model when just a few sites have to be considered (say $n \approx 100$), it becomes hopeless when $n \geq 1000$ because of the huge number of variables and constraints involved, which grows quadratically with n. Therefore, when facing instances with several thousand of sites an alternative (possibly weaker) model is required, where interference can be handled by a number of variables and constraints that grows just linearly with n. The model below is a compact reformulation of model (4.1)–(4.6) that follows a recipe of Glover Glover, 1975 that is widely used, e.g., in the Quadratic Assignment Problem Xia and Yuan, 2006; Fischetti et al., 2012. The original objective function (to be maximized), written as

$$\sum_{i \in V} P_i x_i - \sum_{i \in V} \Big(\sum_{j \in V} I_{ij} x_j \Big) x_i \tag{4.11}$$

is restated as

$$\sum_{i \in V} (P_i x_i - w_i) \tag{4.12}$$

where

$$w_i := \Big(\sum_{j \in V} I_{ij} x_j \Big) x_i = \begin{cases} \sum_{j \in V} I_{ij} x_j & \text{if } x_i = 1; \\ 0 & \text{if } x_i = 0. \end{cases}$$

denotes the total interference caused by site i. Our compact model then reads

$$\max z = \sum_{i \in V} (P_i x_i - w_i) \tag{4.13}$$

$$\text{s.t.} \quad N_{MIN} \leq \sum_{i \in V} x_i \leq N_{MAX} \tag{4.14}$$

$$x_i + x_j \leq 1 \qquad \forall [i,j] \in E_I \tag{4.15}$$

$$\sum_{j \in V} I_{ij} x_j \leq w_i + M_i (1 - x_i) \qquad \forall i \in V \tag{4.16}$$

$$x_i \in \{0,1\} \qquad \forall i \in V \tag{4.17}$$

$$w_i \geq 0 \qquad \forall i \in V \tag{4.18}$$

where the big-M term $M_i = \sum_{\substack{j \in V \\ [i,j] \notin E_I}} I_{ij}$ is used to deactivate constraint (4.16) in case $x_i = 0$.

Our preliminary tests suggested not to explicitly strengthen constraints (4.15) to their clique form (4.7), as a family of cliques is automatically generated during preprocessing by the MIP solver in a very efficient way.

4.3 Which ad-hoc heuristic?

A simple 1- and 2-opt heuristic with local-minimum escape through fictitious bounds on the turbine number was implemented. Other simple heuristics (including tabu search) have been tried but seem to have a worse performance, at least in our implementation.

The core of our heuristic is a parametrized 1-opt search. At each step, we have an incumbent solution, say $\tilde{x}$, that describes the best-known turbine allocation ($\tilde{x}_i = 1$ if a turbine is built at site i, 0 otherwise), and a current solution x. Let

$$z = \sum_{i \in V} P_i x_i - \sum_{i \in V} \sum_{j \in V} I_{ij}\, x_i\, x_j$$

be the profit of the current solution,

$$\gamma = \sum_{i \in V} x_i$$

be its cardinality, and define for each $j \in V$ the extra-profit δ_j incurred when flipping x_j, namely:

$$\delta_j = \begin{cases} P_j - \displaystyle\sum_{i \in V : x_i = 1} (I_{ij} + I_{ji}) & \text{if } x_j = 0; \\ -P_j + \displaystyle\sum_{i \in V : x_i = 1} (I_{ij} + I_{ji}) & \text{if } x_j = 1 \end{cases}$$

where we assume $I_{ij} = BIG$ for all incompatible pairs $[i, j] \in E_I$, and BIG is a large penalty value (e.g., $BIG > \sum_{i \in V} P_i$), while $I_{ii} = 0$ as usual.

We start with $x = 0$, $z = 0$, $\gamma = 0$ and initialize $\delta_j = P_j$ for all $j \in V$. We also define a local copy of N_{MIN} and N_{MAX}, say n_1 and n_2. Then, we iteratively improve x by a sequence of 1-opt moves, according to the following scheme. At each iteration, we look in $O(n)$ time for the site j with maximum $\delta_j + FLIP(j)$,

where function $FLIP(j)$ takes cardinality constraints into account, namely

$$FLIP(j) = \begin{cases} -HUGE & \text{if } x_j = 0 \text{ and } \gamma \geq n_2 \\ -HUGE & \text{if } x_j = 1 \text{ and } \gamma \leq n_1 \\ +HUGE & \text{if } x_j = 0 \text{ and } \gamma < n_1 \\ +HUGE & \text{if } x_j = 1 \text{ and } \gamma > n_2 \\ 0 & \text{otherwise} \end{cases}$$

with $HUGE >> BIG$ (recall that the function $\delta_j + FLIP_j$ has to be maximized). In our implementation we used $BIG = 10,000$ and $HUGE = 1,000,000$. Once the best j has been found, say $j = j^*$, if $\delta_{j^*} + FLIP(j^*) > 0$ we just flip x_{j^*}, update x, z, and γ in $O(1)$ time, update all δ_j's in $O(n)$ time (as explained below), and repeat. In this way a sequence of improving solutions x (and hence $\tilde{x}$) is obtained, a local optimal solution x that cannot be improved by just one flip is found.

As to time complexity, the most time consuming step is the update of each δ_j as a result of the flip of a single x_{j^*}. However, each update requires just $O(1)$ time through the following parametrized formula, to be applied *before* the flip of x_{j^*}:

$$\delta_j = \begin{cases} -\delta_j & \text{if } j = j^* \\ \delta_j - (I_{jj^*} + I_{j^*j}) & \text{if } j \neq j^* \text{ and } x_j = x_{j^*} \\ \delta_j + (I_{jj^*} + I_{j^*j}) & \text{if } j \neq j^* \text{ and } x_j \neq x_{j^*} \end{cases}$$

Validity of the above formula is obvious for $j = j^*$, whereas for the other cases it follows directly from the definition of δ_j by considering the four combinations $(x_j, x_{j^*}) \in \{(0,0),(0,1),(1,0),(1,1)\}$. It then follows that each 1-opt iteration requires $O(n)$ time, as claimed, whereas a non-parametric implementation would require $O(n^2)$ time.

To escape local minima, a number of metaheuristic approaches can be used, e.g., Tabu Search Glover, 1990 or Variable Neighborhhod Search Mladenovic and Hansen, 1997. In our implementation, we used an alternative scheme that produced good results for our instances. The idea is to modify the local limits n_1 and n_2 to force the current x to move to fulfill them, thus visiting different parts of the solution space. More specifically, as soon as we get to a local minimum (i.e., $\delta_{j^*} + FLIP(j^*) \leq 0$), we generate a uniformly pseudo-random value $\rho \in [0,1]$ and update the local limits as follows:

$$n_1 := n_2 := \begin{cases} \gamma(1+\rho/2) + 10 & \text{if } \gamma \leq \sum_{i \in V} \tilde{x}_i \\ \gamma(1-\rho/2) - 10 & \text{otherwise} \end{cases}$$

In this way we obtain an oscillatory behavior where the cardinality of the current x (namely, γ) goes up and down, following a zig-zag trajectory. Each time a new solution x is constructed, the incumbent $\tilde{x}$ is possibly updated by considering the true limits N_{MIN} and N_{MAX} (instead of their local counterparts n_1 and n_2).

In our algorithm, we also apply a sequence of improving 2-opt exchanges to the current solution x, until no improving 2-opt exchange exists. This step is useful as it allows, e.g., to move a single turbine to a nearby (better) site. As each 2-opt exchange requires $O(n^2)$ time, however, this phase is applied in a conservative way, also because it interferes with the zig-zag mechanism and tends to produce worse solutions in the long run. In our implementation, improving 2-opt exchanges on x are only applied immediately before a change of the local limits n_1 and n_2, and on the final incumbent $\tilde{x}$, just before it is returned.

The above heuristic is applied in two different modes. In the "initial solution" mode, we start with $\tilde{x} := x := 0$ and repeat the procedure until we count a very large number (10,000) of consecutive 1-opt calls with no improvement of $\tilde{x}$. In the faster "clean-up" mode, instead, we already have an incumbent $\tilde{x}$ to refine, so we initialize $x := \tilde{x}$ and repeat the procedure until we count 100 consecutive 1-opt calls with no improvement of $\tilde{x}$. As already mentioned, 2-opt exchanges are applied in all cases before the final $\tilde{x}$ is returned.

4.4 Which MIP heuristic?

We now address how to improve a given feasible solution $(\tilde{x}, \tilde{w})$ by exploiting MIP model (4.13)–(4.18). One standard option would be to just use $(\tilde{x}, \tilde{w})$ to initialize the incumbent solution of the MIP solver, and to run it in its default mode. However, it is common experience that this strategy is unlikely to produce improved solutions within acceptable computing times, especially if the underlying MIP model is very large and the formulation is weak—as it happens in out context. So, we preferred to address a different use of the MIP solver, to be applied to "search a neighborhood" of $(\tilde{x}, \tilde{w})$. In particular, our algorithm belongs to the *Large Neighborhood Search* scheme (see, e.g., Shaw, 1998, Focacci et al., 2003 and Pisinger and Ropke, 2010), as we consider an exponentially large neighborhood and explore it using the *proximity search* strategy recently proposed in Fischetti and Monaci, 2014, that seems particularly suited for models involving big-M constraints.

Proximity search works in stages, each aimed at producing an improved feasible solution, and is illustrated in Figure 4.2. At each stage, an explicit cutoff constraint

$$\sum_{i \in V}(P_i x_i - w_i) \geq \sum_{i \in V}(P_i \tilde{x}_i - \tilde{w}_i) + \theta \tag{4.19}$$

is added to the original MIP, where $\theta > 0$ is a given tolerance that specifies the minimum improvement required. The objective function of the problem can then be replaced by a new "proximity function" (to be minimized):

$$\Delta(x, \tilde{x}) = \sum_{j \in V:\, \tilde{x}_j = 0} x_j + \sum_{j \in V:\, \tilde{x}_j = 1} (1 - x_j) \tag{4.20}$$

that measures the Hamming distance between a generic binary x and the given $\tilde{x}$; note that continuous variables w_i's play no role in this definition. One then applies the MIP solver, as a black box, to the modified problem in the hope of finding an improved solution at a small Hamming distance from $\tilde{x}$. The computational experience reported in Fischetti and Monaci, 2014 confirms that this approach is quite successful (at least, on some classes of problems), due to the action of the proximity objective function that is beneficial both in speeding up the solution of the LP relaxations, and in driving the heuristics embedded in the MIP solvers—thus resulting into an improved "relaxation grip" Fischetti and Monaci, 2014.

In our implementation, we used an improved version of the above scheme, called "proximity search with an incumbent" in Fischetti and Monaci, 2014. The idea is that one would like to provide the MIP-solver an incumbent by using the current solution $(\tilde{x}, \tilde{w})$, which is however infeasible because of the cutoff constraint. So, one can introduce a continuous variable $\xi \geq 0$ and weaken (4.19) to its "soft" version:

$$\sum_{i \in V}(P_i x_i - w_i) \geq \sum_{i \in V}(P_i \tilde{x}_i - \tilde{w}_i) + \theta(1 - \xi) \tag{4.21}$$

while minimizing $\Delta(x, \tilde{x}) + U\xi$ instead of just $\Delta(x, \tilde{x})$, where $U >> 0$ is a very large value with respect to Δ; see again Fischetti and Monaci, 2014 for details.

4.5 The overall approach

As already mentioned, our approach can be cast into the *Large Neighborhood Search* paradigm, and in particular in the *MIP-and-refine* framework recently investigated in Fischetti et al., forthcoming, and works as shown in Figure 4.3.

Figure 4.2: The basic proximity search scheme

Proximity search:

let $(\tilde{x}, \tilde{w})$ be the initial feasible solution to improve;

repeat

explicitly add cutoff constraint (4.19) to the MIP model;

install the new "proximity" objective function (4.20) to be minimized;

run the MIP solver on the new model until a termination condition is reached, and let (x^*, w^*) be the best feasible solution found;

refine w^* by solving the original model (4.13)–(4.18) after fixing $x = x^*$;

recenter $\Delta(x, \cdot)$ by setting $\tilde{x} := x^*$, and/or update θ

until *an overall termination condition is reached*;

return $(\tilde{x}, \tilde{w})$

Figure 4.3: Our overall heuristic framework

Step 0. read input data and compute the overall interference matrix (I_{ij});

Step 1. apply ad-hoc heuristics (iterated 1-opt) to get a first incumbent $\tilde{x}$;

Step 2. apply quick ad-hoc refinement heuristics (few iterations of iterated 1- and 2-opt) to possibly improve $\tilde{x}$;

Step 3. if $n > 2000$, randomly remove points $i \in V$ with $\tilde{x}_i = 0$ so as to reduce the number of candidate sites to 2000;

Step 4. build a MIP model for the resulting subproblem and apply proximity search to refine $\tilde{x}$ until the very first improved solution is found (or time limit is reached);

Step 5. if time limit permits, repeat from Step 2.

At Step 1. (respectively, Step 2.) the ad-hoc heuristic of Section 4.3 is applied in its initial-solution (resp., clean-up) mode. Two different MIP models are

used to feed the proximity-search heuristic at Step 4. During the first part of the computation, we use a simplified MIP model obtained from (4.13)–(4.18) by removing all interference constraints (4.16), thus obtaining a much easier relaxation. A short time limit (60 sec.s) is imposed for each call of proximity search when this simplified model is solved. In this way we aggressively drive the solution $\tilde{x}$ to increase the number of built turbines, without being bothered by interference considerations and only taking pairwise incompatibility (4.15) into account. This approach quickly finds better and better solutions (even in terms of the true profit), until either (i) no additional turbine can be built, or (ii) the addition of new turbines does in fact reduce the true profit associated to the new solution. In this situation we switch to the complete model (4.13)–(4.18) with all interference constraints, which is used in all next executions of Step 4. Note that the simplified model is only used at Step 4, while all other steps of the procedure always use the true objective function that takes interference into full account.

4.6 Computational results

The following alternative solution approaches were implemented in C language, some of which use the commercial MIP solver IBM ILOG Cplex 12.5.1 IBM ILOG CPLEX, 2013; because of the big-M's involved in the models, all Cplex's codes use zero as integrality tolerance (`CPX_PARAM_EPINT` = 0.0).

a) `proxy`: our MIP-and-refine heuristic, as outlined in the previous section, using Cplex with the following aggressive parameter tuning: all cuts deactivated, `CPX_PARAM_RINSHEUR` = 1, `CPX_PARAM_POLISHAFTERTIME` = 0.0, `CPX_PARAM_INTSOLLIM` = 2;

b) `cpx_def`: the application of IBM ILOG Cplex 12.5.1 in its default setting, starting from the same heuristic solution $\tilde{x}$ found by `proxy` after the first execution of Step 2 of Figure 4.3;

c) `cpx_heu`: same as `cpx_def`, with the following internal tuning intended to improve Cplex's heuristic performance: all cuts deactivated, `CPX_PARAM_RINSHEUR` = 100, `CPX_PARAM_POLISHAFTERTIME` = 20% of the total time limit;

d) `loc_sea`: a simple local-search procedure not based on any MIP solver, that just loops on Steps 2 of Figure 4.3 and randomly removes installed turbines from the current best solution after 10,000 iterations without improvement of the incumbent.

For each algorithm, we considered the best solution found within a given time limit. In our view, `loc_sea` is representative of a clever but not oversophisticated metaheuristic, as typically implemented in practice, while `cpx_def` and `cpx_heu` represent a standard way of exploiting a MIP model once a good feasible solution is known.

Our testbed refers to an offshore 3,000 × 3,000 (m) square with $D_{MIN} = 400$ (m) minimum turbine separation, with no limit on the number of turbines to be built (i.e., $N_{MIN} = 0$ and $N_{MAX} = +\infty$). Turbines are all of Siemens SWT-2.3-93 type (diameter 93m), which produces a power of 0.0 MW for wind speed up to 3 m/s, of 2.3 MW for wind speed greater than or equal to 16 m/s, and intermediate values for winds in range 3-16 m/s according to a nonlinear function Siemens AG, n.d. Pairwise interference (in MW) was computed using Jensen's model Jensen, 1983, by averaging 250,000+ real-world wind samples. Those samples were grouped into about 500 macro-scenarios to reduce the computational time spent defining the interference matrix. A pairwise average interference of 0.01 MW or less is treated as zero. The reader is referred to Fischetti, 2014 for details.

We generated five classes of medium-to-large problems with $n = 1000$, 5000, 10000, 15000, and 20000. For each class, 10 instances have been considered by generating n uniformly random points in the 3,000 × 3,000 square. (Although in the offshore case turbine positions are typically sampled on a regular grid, we decided to randomly generate them to be able to compute meaningful statistics for each value of n.)

In what follows, reported computing times are in CPU seconds of an Intel Xeon E3-1220 V2 quad-core PC with 16GB of RAM, and do not take Step 0 of Figure 4.3 into account as the interference matrix is assumed to be precomputed and reused at each what-if analysis run.

Computational results on our instances are given in Table 4.1, where each entry refers to the performance of a given algorithm at a given time limit. In particular, the left part of the table reports, for each algorithm and time limit, the *number of wins*, i.e, the number of instances for which a certain algorithm produced the best solution at the given time limit (ties allowed).

According to the table, `proxy` outperforms all competitors by a large amount for medium to large instances. As expected, `cpx_heu` performs better for instances with $n = 1{,}000$ as it is allowed to explore a large number of enumeration nodes for the original model and objective function. Note that `loc_sea` has a good performance for short time limits and/or for large instances, thus confirming

its effectiveness, whereas `cpx_heu` is significantly better than `loc_sea` only for small instances and large time limits.

A different performance measure is given in the right-hand side part of Table 4.1, where each entry gives the *average optimality ratio*, i.e., the average value of the ratio between the solution produced by an algorithm (on a given instance at a given time limit) and the best solution known for that instance—the closer to one the better. It should be observed that an improvement of just 1% has a very significant economical impact due to the very large profits involved in the wind farm context. The results show that `proxy` is always able to produce solutions that are quite close to the best one. As before, `loc_sea` is competitive for large instances when a very small computing time is allowed, whereas `cpx_def` and `cpx_heu` exhibit a good performance only for small instances, and are dominated even by `loc_sea` for larger ones.

Figure 4.4 plots the incumbent value (i.e., the profit of the current best solution) over CPU time for the four heuristics under comparison, and refer to 4 sample instances in our testbed. The two subfigures on the top refer to two small instances with $n = 1{,}000$, where `proxy`, `cpx_heu` and `cpx_def` have a comparable performance and clearly outperform `loc_sea`. For $n = 5{,}000$ (bottom-left subfigure) and $n = 10{,}000$ (bottom-right subfigure), however, both `cpx_def` and `cpx_heu` (and also `loc_sea`) have hard time in improving their initial solution, and are outperformed by `proxy` by a large amount.

4.7 Conclusions

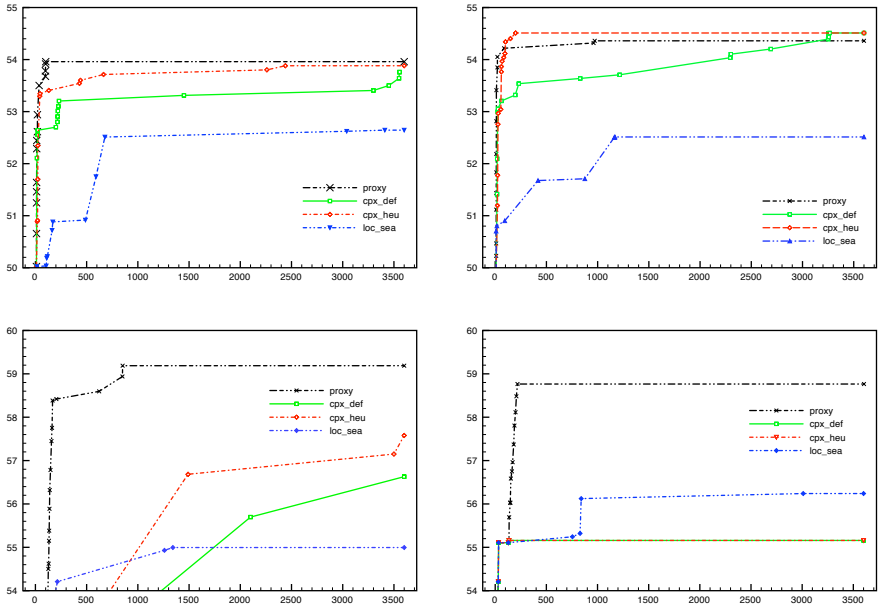
We have considered an important practical problem in wind farm optimization, namely, the optimal allocation of turbines subject to interference conditions. Our goal was the design of a fast heuristic capable of handling instances with 10,000+ potential sites in a matter a minutes. To this end, we have exploited two basic tools: a fast ad-hoc heuristic, and a MIP model designed for the very large instances of interest. A synergic use of these two tools has been proposed, following a clever MIP-and-refine recipe where two different variants of the underlying MIP model have been solved through a proximity search heuristic. Computational results on a testbed of medium-to-large scale instances have shown that the approach outperforms a standard use of the two basic tools.

A lesson learned is that the choice of the MIP model to be used is a critical step in the design of the overall heuristic framework, because an effective compromise between tightness and compactness is required. In particular, models

Table 4.1: Number of times each algorithm finds the best solution within the time limit (wins), and optimality ratio with respect to the best known solution—the larger the better.

		number of wins				optimality ratio			
n	Time limit (s)	proxy	cpx_def	cpx_heu	loc_sea	proxy	cpx_def	cpx_heu	loc_sea
1,000	60	6	1	3	0	0.994	0.983	0.987	0.916
	300	4	2	4	0	0.997	0.991	0.998	0.922
	600	7	3	7	0	0.997	0.992	0.997	0.932
	900	5	2	3	0	0.998	0.993	0.996	0.935
	1,200	5	1	5	0	0.998	0.992	0.997	0.939
	1,800	5	1	4	0	0.998	0.992	0.996	0.942
	3,600	4	2	5	0	0.998	0.995	0.997	0.943
5,000	60	9	6	6	5	0.909	0.901	0.901	0.904
	300	10	0	0	0	0.992	0.908	0.908	0.925
	600	10	0	10	0	0.994	0.908	0.994	0.935
	900	10	0	0	0	0.994	0.908	0.908	0.936
	1,200	10	0	0	0	0.994	0.908	0.925	0.939
	1,800	9	0	1	0	0.996	0.908	0.971	0.946
	3,600	5	0	5	0	0.996	0.932	0.994	0.948
10,000	60	9	9	8	10	0.914	0.913	0.914	0.914
	300	10	2	2	2	0.967	0.927	0.927	0.936
	600	10	0	10	0	0.998	0.928	0.998	0.944
	900	10	0	0	0	1.000	0.928	0.928	0.948
	1,200	10	0	0	0	1.000	0.928	0.928	0.951
	1,800	10	0	0	0	1.000	0.928	0.928	0.957
	3,600	9	0	0	1	1.000	0.928	0.928	0.964
15,000	60	9	10	9	9	0.909	0.912	0.911	0.909
	300	10	8	7	8	0.943	0.937	0.935	0.937
	600	10	0	10	0	0.992	0.939	0.992	0.942
	900	10	0	0	0	1.000	0.939	0.939	0.956
	1,200	9	0	0	1	1.000	0.939	0.939	0.959
	1,800	9	0	0	1	1.000	0.939	0.939	0.965
	3,600	9	0	0	1	1.000	0.939	0.939	0.972
20,000	60	9	9	9	10	0.901	0.902	0.901	0.902
	300	10	8	10	10	0.933	0.933	0.933	0.933
	600	9	0	9	1	0.956	0.935	0.956	0.941
	900	10	0	0	0	0.978	0.935	0.935	0.945
	1,200	10	0	0	0	0.991	0.935	0.935	0.950
	1,800	10	0	0	0	0.999	0.935	0.935	0.963
	3,600	9	0	0	0	1.000	0.935	0.935	0.971
ALL	60	42	35	35	34	0.925	0.922	0.922	0.909
	300	44	20	23	20	0.966	0.939	0.940	0.930
	600	46	3	46	1	0.987	0.941	0.987	0.938
	900	45	2	3	0	0.994	0.941	0.941	0.944
	1,200	44	1	5	1	0.997	0.940	0.945	0.947
	1,800	43	1	5	1	0.999	0.940	0.954	0.955
	3,600	36	2	10	2	0.999	0.946	0.959	0.959

Figure 4.4: Solution profit over time for 4 sample instances with n = 1,000 (top left and top right), n = 5000 (bottom left), and n = 10,000 (bottom right); the higher the profit the better.



that are considered weak when solving small instances to proven optimality can become effective when used in a refining mode for large instances. In addition, simplified MIP models that relax some details of the problem (the effect of interference, in our case) can be very useful at the early stage of the heuristic.

Future research should evaluate different ways to sparsify the problem by removing candidate sites (Step 3 of Figure 4.3). In our runs we used a simple random criterion, but more clever options that favor the removal of points far from all installed turbines are also possible. By putting this mechanism to its extreme extent, it is in fact conceivable to address a "continuous" version of the problem where a turbine can be installed at any points in a certain geographical area, and the heuristic dynamically discretizes it by generating and removing sites $i \in V$ on the fly.

Acknowledgements

The research of the second author was supported by Miur (project PRIN 2012) and by the University of Padova (Progetto di Ateneo "Exploiting randomness in Mixed Integer Linear Programming"). We thank Jakob Stoustrup and John Leth (Automation and Control, Department of Electronic Systems, Aalborg University) who supervised the thesis work of first author in Aalborg, and Benjamin Martinez (R&D Wind Engineer, Vattenfall AB) who provided the real-world wind scenarios used in our tests.

References

Archer, R., G. Nates, S. Donovan, and H. Waterer (2011). "Wind Turbine Interference in a Wind Farm Layout Optimization Mixed Integer Linear Programming Model". In: *Wind Engineering* 35 (2), pp. 165–178.

Barthelmie, R., K. Hansen, S. T. Frandsen, O. Rathmann, J. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. Politis, and P. K. Chaviaropoulos (2009). "Modelling and measuring flow and wind turbine wakes in large wind farms offshore". In: *Wind Energy* 12, pp. 431–444.

Donovan, S. (2005). "Wind farm optimization". In: *Proceedings of the 40th Annual ORSNZ Conference*, pp. 196–205.

EMD (n.d.). *WindPRO*. `http://www.emd.dk/windpro/frontpage`.

Fischetti, M. (2014). "Mixed-Integer Models and Algorithms for Wind Farm Layout Optimization". `http://tesi.cab.unipd.it/45458/1/tesi_Fischetti.pdf`. MA thesis. University of Padova.

Fischetti, M. and M. Monaci (2014). "Proximity search for 0-1 mixed-integer convex programming". In: *Journal of Heuristics* 6.20, pp. 709–731.

Fischetti, M., M. Monaci, and D. Salvagnin (2012). "Three Ideas for the Quadratic Assignment Problem". In: *Operations Research* 60.4, pp. 954–964.

Fischetti, M., G. Sartor, and A. Zanette (forthcoming). "A MIP-and-refine matheuristic for smart grid energy management". In: *International Transactions in Operational Research*. DOI:10.1111/itor.12034.

Fischetti, M. and M. Monaci (2016). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: `10.1007/s10732-015-9283-4`. URL: `https://doi.org/10.1007/s10732-015-9283-4`.

Focacci, F., F. Laburthe, and A. Lodi (2003). "Local Search and Constraint Programming". In: *Handbook of Metaheuristics Management Science*. Vol. 57, pp. 369–403.

Glover, F. (1975). "Improved Linear Integer Programming Formulations of Nonlinear Integer Problems". In: *Management Science* 22 (4), pp. 455–460.

Glover, F. (1990). "Tabu Search: A Tutorial". In: *Interfaces* 20.4, pp. 74–94.

IBM ILOG CPLEX (2013). *Optimization Studio*. `http://www.cplex.com`.

Jensen, N. (1983). *A note on wind generator interaction*. Tech. rep. Technical Report Riso-M-2411(EN), Riso National Laboratory, Roskilde, Denmark.

Kusiak, A. and Z. Song (2010). "Design of wind farm layout for maximum wind energy capture". In: *Renewable Energy* 35, pp. 685–694.

Mladenovic, N. and P. Hansen (1997). "Variable neighborhood search". In: *Computers & Operations Research* 24.11, pp. 1097–1100.

Pisinger, D. and S. Ropke (2010). "Large Neighborhood Search". In: *Handbook of Metaheuristics, International Series in Operations Research & Management Science*. Vol. 146, pp. 399–419.

Renkema, D. (2007). "Validation of wind turbine wake models". MA thesis. Delft University of Technology.

Shaw, P. (1998). "Using constraint programming and local search methods to solve vehicle routing problems". In: *4th International Conference CP98, Lecture Notes in Computes Science 1520.* Pp. 417–431.

Siemens AG (n.d.). *SWT-2.3-93 Turbine, Technical Specifications*. `http://www.energy.siemens.com`.

Vattenfall (2014). *Personal communication*.

Xia, Y. and Y. Yuan (2006). "A new linearization method for quadratic assignment problem". In: *Optimization Methods and Software* 21, pp. 803–816.

Chapter 5

Wind Farm Layout Optimization For Multiple Turbine Types and Multiple Zones

Publication Status: Work partially presented at an International Conference (INFORMS 2016)

Reading Instructions: Extension of the Fischetti and Monaci, 2016 paper (read after Chapter 4)

5.1 Introduction

In this chapter we extend the models from Fischetti and Monaci, 2016 (in Chapter 4) to consider multiple turbine types. Reasons for considering multiple

turbines are mainly of three types:

1. two (or more) turbine types can be mixed in the same site;
2. the whole area available is divided in zones; the layout can cover all zones, but in each zone a specific turbine type must be used;
3. the layout to be optimized consists of only one turbine type, but it needs to consider the interference from surrounding parks (that may consist of different turbine types).

The first option could be used to explore the potential of having different turbines from different manufacturers in the same site. Even if this option is feasible in theory, our company partner suggested that it is not convenient, since it implies having different contracts to buy turbines, as well as to maintain them during the park lifetime. This may result in high extra costs. In our tests, we therefore considered a different option for point 1, namely considering variations of the same turbine from the same manufacturer. In particular, in Subsection 5.3.1 we consider three variants of the same turbine model, that differ only in rotor diameter. Each turbine variant is considered as a turbine type, and the optimizer is free to mix them if convenient. While point 1 (mixing turbines in the same layout) may be considered as a research case (as the company does not mix turbines nowadays), point 2 and 3 actually arise in current wind parks.

The reasons for dividing a site in zones may be various. In Subsection 5.3.2 we consider a noise limitation reason: offshore sites located close to shore may be subdivided in different zones, where noise limitations apply on the closest zones to shore. In those zones, to reduce noise emissions the turbines can operate only at reduced speed, i.e., they can operate only if curtailed. The standard version of the turbine and its curtailed version then represent different turbine types in our optimization. The optimizer is free to locate turbines wherever in the whole area, but is limited to use only curtailed turbines in the near-to-shore one.

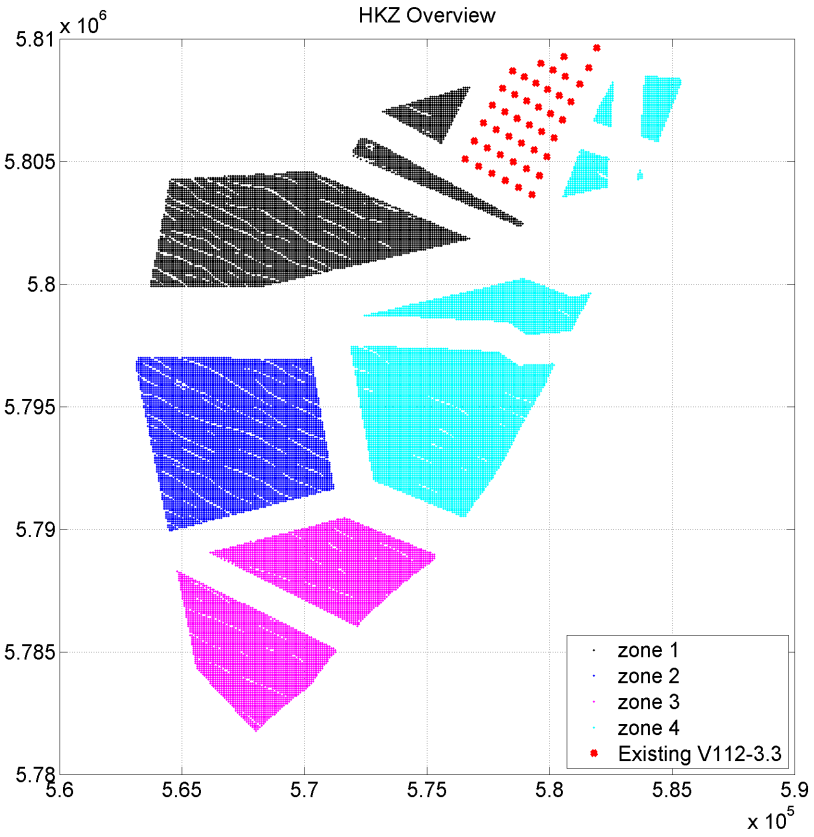
Another reason to divide the full area in zones (still point 2), refers directly to the tender rules. Very big sites (as Danish Kriegers Flak, or Borssele; see Part IV) may be divided in smaller zones to allow wind park constructors to bid only for some of the zones. This opens up the competition also to smaller players, that may not be able to participate for the full site, but can be interested in competing for a part of it. It can happen, then, that a big area is divided in, e.g., two zones and the developer company can enter the tender only for one of the sites, or delivering a combined bid for both sites. In the second case,

it is important to optimize simultaneously the layouts, as they may interfere one with the other, causing revenue losses in the overall project. Nevertheless, each of the site may have its specific limitations (namely, a different maximum number of turbines) and different turbine types may potentially be used (one type for each site). In this case, point 2 normally has the additional limitation of having a fixed number of turbines for each site. In Chapter 10 we address this application in detail, for a real wind park (namely, Danish Kriegers Flak).

Point 3 (considering existing parks nearby), is very often used in real-world park design. When a new site is available to construct a park, it may be that other wind parks already exist nearby. These existing parks cannot, of course, be changed, but their turbines may interfere with those of the new park. It is therefore very important to consider wake effects from nearby parks while optimizing the new layout. These already existing parks typically have different turbines models than the one of the newly designed layout, because companies bid using the newest turbine models (more efficient and powerful), and the existing parks have older turbine models. Therefore different turbine types must be considered in the interference evaluation. Notice that the existing parks may be owned by competitors, so we do not care if our layout interfere with the existing parks but, on the contrary, we focus only on the "received" interference.

A real-world example of a multi-zone application with existing parks nearby is Hollandse Kust Zuid Holland wind park project. The whole area was divided in four sites, where sites I and II were on tender in 2017, and sites III and IV in 2018. Figure 5.1 shows the different areas open for tender with different colours. In the north part of the site there is an already existing park, Luchterduinen. The position of Luchterduinen turbines are fixed, and they are indicated as red dots in the plot of Figure 5.1. This is a concrete example in which the optimizer should consider both different zones and existing parks (points 2 and 3 of our application list, together).

Figure 5.1: Hollandse Kust Zuid Holland is divided in 4 sites (in different colours in the plot) and suffers the interference from an existing park (red dots)



5.2 Modelling multiple turbine types

We will now address the potential of considering different types of turbine in our wind farm layout optimization. First, we will focus to the first application we named in Section 5.1 (i.e. mixing different turbine types in the same area). The other applications may be considered by slightly modifying this option, so we will look at them in a second stage.

So let us say that we have T different types of turbines available, that we can freely use in our optimization. In order to do that, one could change the model by defining a new variable

$$x_i^t = \begin{cases} 1 & \text{if a turbine of type } t \text{ is built at position } i \in V; \\ 0 & \text{otherwise} \end{cases} \qquad i \in V, t = 1, \ldots, T \tag{5.1}$$

where $t = 1, \ldots, T$ is the turbine type. Notice that, in practical applications, T is expected to be small (two or three kinds of turbines to compare). Nevertheless one could perform the same optimization without explicitly introducing a new variable x_i^t, but simply by cloning the input points T-times and by labling each of the resulting points with a specific $t = 1, \ldots, T$. Due to the proximity constraint in the model, the optimization will never choose to build two turbines one on top of the other, therefore the optimizer is implicitly forced to select a unique turbine type for each position. The code is adapted so that it computes the wake interference based on the type of the selected turbine (see Appendix A, Section A.4 for details). As to the minimum distance between turbines, we always refer to the bigger rotor diameter between all the turbine types available.

Since different kinds of turbines can potentially have a different rated power (i.e., maximum power production) we had to rewrite (4.14) so that we do not refer anymore to a maximum number of turbines, but instead to a maximum power production for the entire site. This approach actually mimics how the maximum number of turbines is defined by the project team. By tender rules, indeed, a maximum MW production for the site must be ensured. If only one turbine type may be used is the site, it is enough to divide this value by the rated power (i.e., the maximum production) of the selected turbine model. This is why we always considered a maximum number of turbines so far. When mixing different turbines with different rated powers, it is easier to handle this constraint in its original form and impose a maximum rated power for all the park. Let us define by R_i the rated power for a turbine at position i, and MAX_POW the power rating allowed for all the wind farm. Note that, by the definition of the problem, R_i should also depend on the turbine type, but using the trick of cloning the points, we can handle the type implicitly. So the constraint reads:

$$\sum_{i \in V} R_i x_i \quad \leq \quad MAX\_POW. \tag{5.2}$$

Different types of turbine normally also have different prices. Of course, it is important to consider this in the optimization, as it could impact the turbine selection. The cost for each turbine type is an input data and can be defined depending on the business case: this price should for sure include the price of

the turbine itself, but could also include other factors related to the turbine model (for example, an extra operational or maintenance cost per turbine type, component costs, etc.). Still using the cloning trick, we can associate a cost c_i for each possible position ($i \in V$), such that point i labeled as type t would have associated the costs related to turbine type t. In order to consider this cost in the optimization, we have to change the objective function (4.13) as follows:

$$\max z = \sum_{i \in V}((P_i - \frac{c_i}{K_{euro}})x_i - w_i) \tag{5.3}$$

where K_{euro} is a discounting factor to compare a cost (in €) with a power production (in MW). This K_{euro} is defined has the net present value of 1 MW production over the lifetime of a turbine. This value is computed by company's experts considering interests, market prices, subsidies and expected lifetime of the wind turbine (typically, 20 years), and it is site specific.

So far, we considered the case where different turbine types may be freely mixed in a site. As already discussed, it may be the case that only a specific turbine type can be used in a specific area (see, for example, the noise application of Subsection 5.3.2). In this case, one should pay attention that cloning is performed accordingly. The cloning strategy, indeed, should take care of the differences in the different zones. In this case, the user should provide on input the zone division (by simply associating a zone z to each possible position i). For each zone z, there will be a different number T_z of turbine models available on input. The cloning function must simply ensure that all points in zone z must be cloned T_z times, and each cloned point must be associated to one of the z turbine models actually available. The constraint relative to the maximum number of turbines (or, more generally, to the maximum power production constraint 5.2) is simply substituted by a set of equivalent constraints:

$$\sum_{i \in V_z} R_i x_i \quad \leq MAX\_POW_z, \quad \forall z \in Z \tag{5.4}$$

where $V_z \in V$ is the set of possible positions in zone z and MAX_POW_z is the maximum production for zone z. Objective function (5.3) stays unchanged.

Let us finally address the case of existing nearby parks. These turbines need to be included in the optimization as they may interfere with the new layout, but their position is fixed. There are two options:

1. the bidding company owns both the new and the existing parks; or

2. the existing park(s) is (are) owned by a competitor.

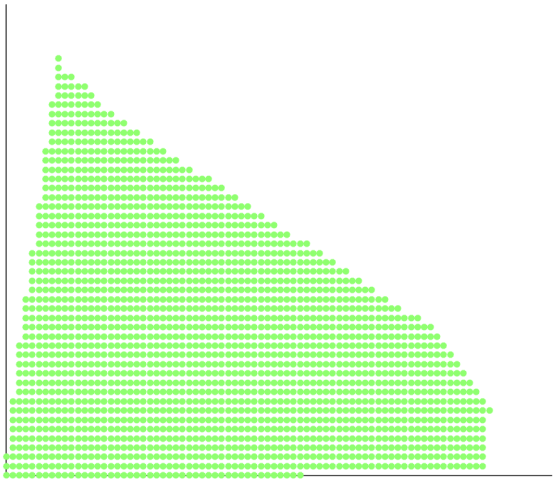
In the first case, we are also interested in the interference that the new park will have on the old park; in the second case we do not.

The set of the fixed positions for each existing park (say, V_{fixed}) is given, together with the turbine model(s). Since the position of the existing turbines is fixed, we simply handle them by fixing to one the variables relative to their position ($x_i = 1 \; \forall i \in V_{fixed}$). Their cost is also fixed to zero, as we have no new costs related to the already existing parks ($c_i = 0 \; \forall i \in V_{fixed}$). If the existing parks are owned by competitors (case 2 above), we may not be interested in the fact that also the existing park may suffer from wake effects from our new park. In this case, indeed, all the production revenue of the existing park go to a competitor, and should therefore not be reflected in our objective. To do so, the pre-computation of the matrix I is modified such that the interference experienced by any $j \in V_{fixed}$ because of any $i \in V$ is artificially set to zero ($I_{i,j} = 0 \; \forall i \in V, \forall j \in V_{fixed}$).
Finally, the production of the existing turbines is artificially set to zero in the objective function (5.3) ($P_i = 0 \; \forall i \in V_{fixed}$).

5.3 Tests

In Section 5.1, we identified three main reasons for mixing turbines in a site. We will here give three examples of application of our optimizer for multiple turbine types, one for each possible reason. The applications are: turbines with the same rated power but different rotor, noise curtailment, and existing nearby parks. For the first two cases, we suppose that we have the artificial site shown Figure 5.2. It consists of 2000 points covering an area of about 5x3 km, regularly sampled on a grid, 75 meters from each point to the next. We considered over 70000 wind scenarios defined by sampling real-world wind time series every 1 m/s (wind speed) and every 0.1 degrees (direction). The time series come from an offshore site in the Netherlands [private communication]. We suppose that the grid operator asks the park to produce 180 MW in total. The type of turbines considered varies from test to test and will be described in the next sections. In these two tests we assume the minimum distance between turbines to be 1 rotor diameter.

Figure 5.2: Artificial site considered for our tests.



For the last application (i.e., considering existing nearby parks in the optimization) we will consider real-world examples. We will look at this application in Subsection 5.3.3.

5.3.1 Different rotor diameters

Nowadays it is not common to mix different turbine models in the same site. Nevertheless, this is an interesting application to investigate when looking for unexploited potential for a site. In this example, in particular, we look at three versions of the same turbine model with different rotor dimensions. The impact of having different rotor diameters is both on the cost of the turbine itself, and on its power curve. A turbine with a bigger rotor can produce more at medium wind speeds but also has a higher initial cost. In order to understand which is the best rotor diameter choice for a specific site, we would like the tool to be able to choose between different kinds of turbines (in this case the same turbine with smaller or bigger rotor). In this test we considered three turbines from Vestas (V116, V117, V126), all with 3.3 MW rated power, but with different rotor diameters (116, 117 and 126 meters). Figure 5.3 shows how the power curves vary depending on the rotor diameter. We estimated the cost of these turbines to be 5.07 M€ for V126, 4.8 M€ for V117 and 4.6 M€ for V116 [1].

[1] Artificial prices. These costs do not refer to any real project

As it can be seen from the power curves, the bigger the rotor the higher the production at medium wind speeds. For example, if the turbine experiences a wind at 8.5 m/s it will produce 2.06 MW if the rotor diameter is 126 m, 1.78 MW if it is 117 m and 1.64 MW if it is 116 m. The optimization tool should then evaluate if it is convenient or not to invest in a bigger rotor and for how many turbines in the site. We assumed $K_{euro} = 0.68$ € (value suggested by our company partner for a generic site). The minimum distance is 126 m in this test.

Figure 5.3: Power curve for Vestas V126 (pink), V117 (red) and V116 (black).

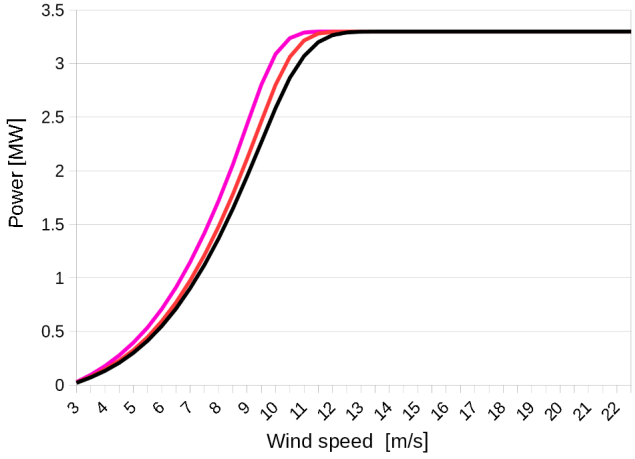
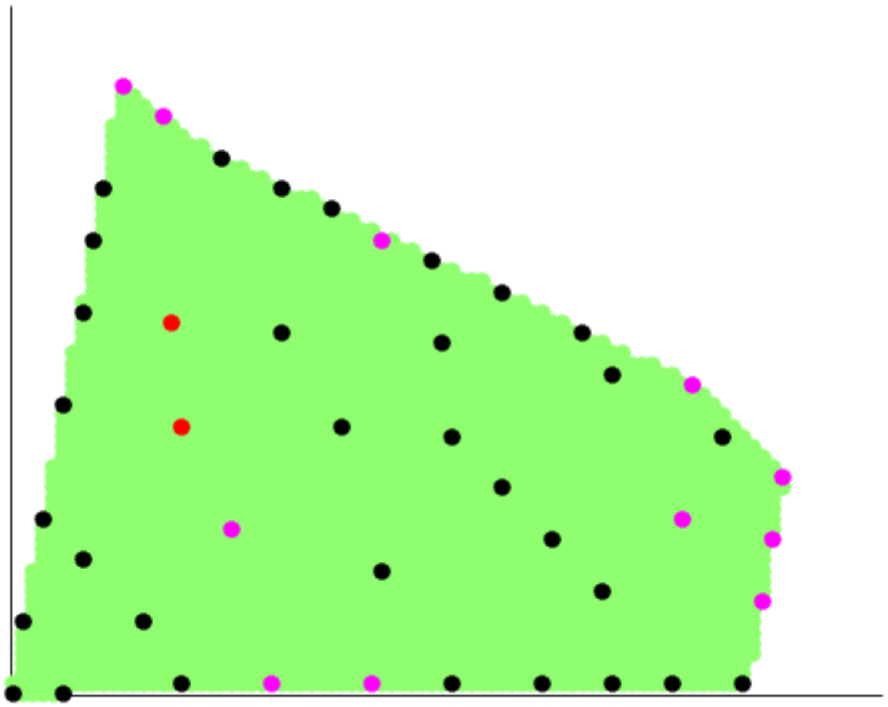


Figure 5.4 shows the optimized layout for our artificial site: each dot represents a turbine and its color indicates the turbine type (Vestas V126 in pink, V117 in red and V116 in black). V116 has a competitive price so it is more often used in the site. This layout of 45 turbines has a cost of value of 212 M€ and a production of about 394 MWh/y.

Figure 5.4: Optimized layout mixing three different turbine types Vestas V126 in pink, V117 in red and V116 in balck.



Notice that if only turbines V116 were used in this site, the optimized layout would have 47 turbines, with a cost of 216 M€ and slightly better production (402 MWh/y).

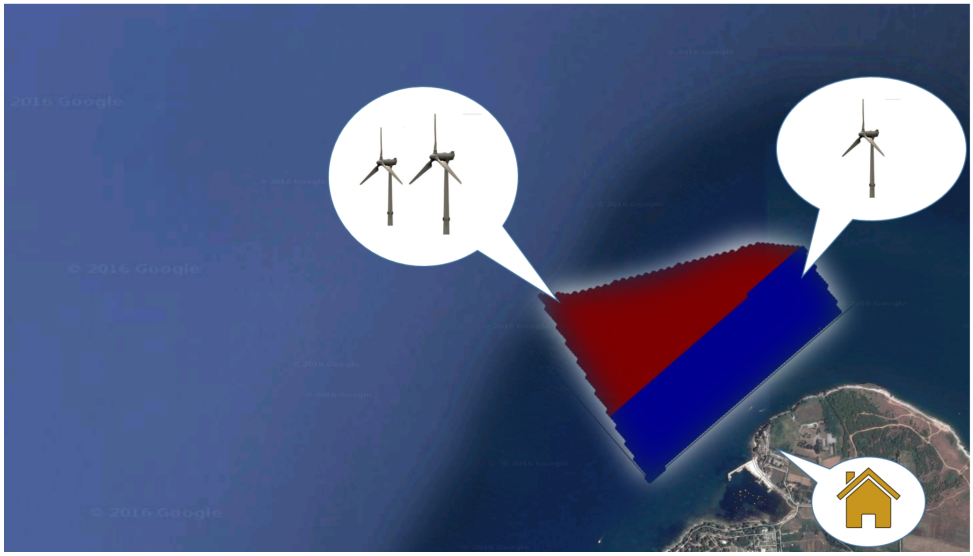
All in all, the two solutions are in this case of comparable value, also due to the similarity between the turbines models, the mixed one being only slightly better.

5.3.2 Application to noise restriction

In this test we consider the same area as before but now we suppose it is located nearshore. On the coast of our artificial site, we have some houses. Noise regulations impose that, until a certain distance from inhabited area, the noise emissions from turbines cannot exceed a certain limit. The wind park designers decided to construct 3.2 MW turbines (rotor diameter 113 m). These turbines, in their standard operation, emit too much noise for the nearby area, therefore turbines located close to shore must be curtailed. This means that the turbines are down-regulated, in order to emit less noise, but, as a consequence, they would also produce less power. Having a total limit of 180 MW (rated power) for the whole park, the optimizer has to balance between locating down-regulated turbines close to shore and pushing all the turbines in the further away area. The second option has the advantage of using the turbines at their full potential, but may imply high wake effects.

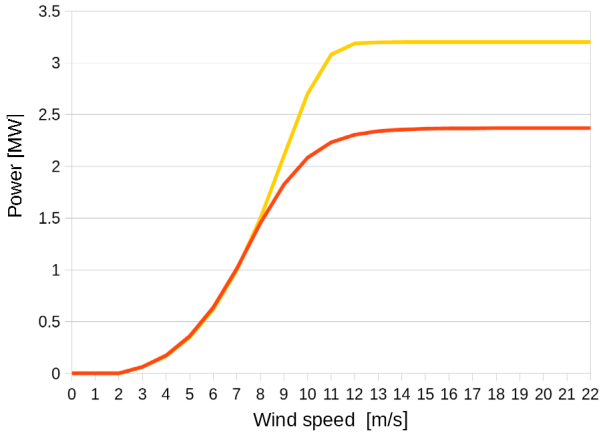
In order to find out how to optimally spread the turbines, we divided the site in 2 areas (blue and red in Figure 5.5): in the area closer to shore (in blue) only curtailed turbines can be located, while in the further away area (in red) both standard and curtailed turbines can be located. We assume the cost of a turbine to be 4 M€, independently of their operation mode (curtailed or standard).

Figure 5.5: Our artificial site is now subject to noise constraints: only curtailed turbines can be placed in the area closer to shore (in blue).



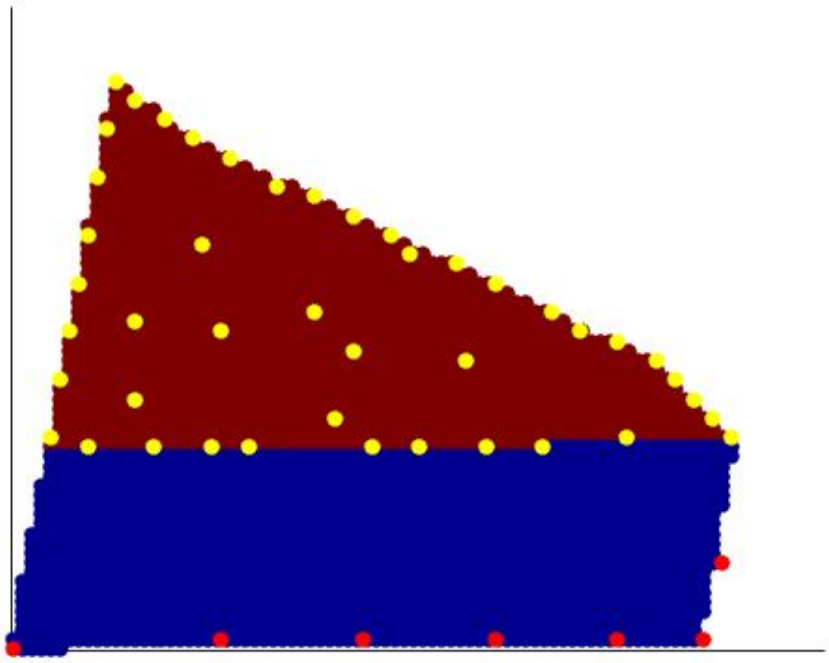
Turbines are curtailed to reduce their noise, but also their production is reduced (in this case from 3.2 to 2.37 MW) as shown in the following plot.

Figure 5.6: A standard 3.2 MW turbine (yellow power curve) is curtailed for noise reason to 2.37 MW (red power curve).



Using the approach introduced in Section 5.2, we therefore cloned all the points in the red area (where both types of turbines could be placed) and ran our optimization. The result is shown in Figure 5.7: the layout balances between the use of curtailed and not curtailed turbines. Curtailed turbines are less convenient as they have the same turbine cost but less production, so only few of them are used in the solution. Nevertheless, using only standard turbines in the further area would result in a higher wake effect.

Figure 5.7: Our artificial site is now subject to noise constraints: only curtailed turbines (in red) can be placed in the area closer to shore (in blue). Yellow dots represent 3.2MW turbines used in their standard mode.



5.3.3 Considering the interference from existing parks

As already mentioned, nowadays it is really common to extend already existing parks, or to construct parks close one to another. In many big offshore parks, it is also common to divide the entire site in zones, that are put on tender in different rounds. This means that different companies could win the tender for the different sub-areas and define their layouts independently. In any case, when constructing a new park close to another one, the interference from the existing park should be considered in the optimization. In general, the existing turbines could be very different from the one selected for the new park. This is because different parks may be owned by different companies, but also because

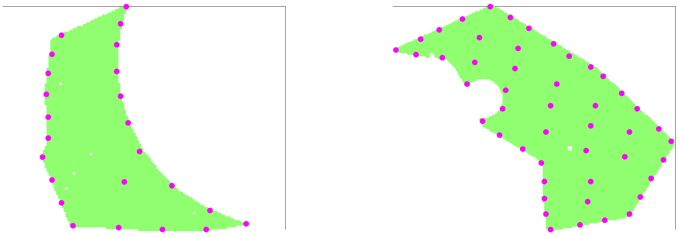
turbine technologies is quickly evolving. Therefore, if we are now constructing the extension of a park constructed, let us say, 5 years ago, we will have now way different (and more powerful) turbines today than then. This means that, when considering nearby parks in the optimization, in most of the cases one has to deal with different turbine types in the area. Note that, as a main difference from the cases above, now the position of some turbine types must be defined by the optimizer, while some other types have fixed positions. According to our experience, normally one wants to optimize a layout with one turbine type, possibly considering many other surrounding sites (with their specific turbine type fixed).

We used Danish Kriegers Flak (DKF) as an example. As shown in Figure 5.8, the whole area is divided in two parts: a East and a West site. A park already exists nearby (red dots in figure): this park is owned by a competitor and may create wake effect on our layout. We assume that one wants to bid on both sites. There is a limitation on the MW production for each site, that equated to a fixed number of turbines to be built in each site. We also assume that the same turbine type in both the sites: 24 turbines in the West site and 48 on the East one.

Figure 5.8: Danish Kriegers Flak is divided in two sites (East and West) and suffers the interference from an existing park (red dots).

Using the model without the extensions for multiple turbines, the only way to handle this was case to optimize the two sites separately. The resulting layout for the whole DKF project is suboptimal, since it does not consider the interference between the West and the East site, nor the one from the existing park nearby. The resulting layouts are shown in Figure 5.9. Their

Figure 5.9: Layout for the two DKF sites optimized separatelly and without considering existing nearby parks.

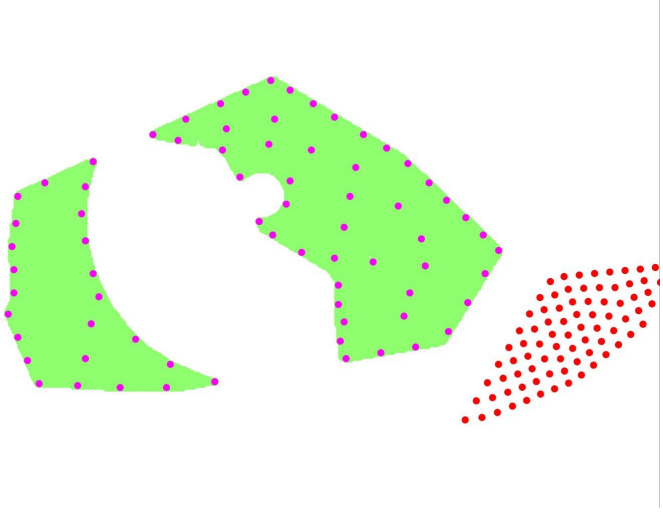


(a) West site *(b) East site*

total production, post-computed considering the mutual interference and the existence of the nearby park, is 4006496 MWh/y.

Thanks to the method of Section 5.2, we can explicitly consider the existing parks and optimize the full DKF project layout while imposing a fixed number of turbines for each site. The resulting layout is shown in Figure 5.10. The power production of this layout is 4008341 MWh/y.

Figure 5.10: Danish Kriegers Flak East and West optimized together, explicitly considering the interference from the existing park nearby.



Comparing the two layouts, the one optimized considering explicitly the mutual interference between sites and the interference from surrounding parks (in Figure 5.10) produces 1845 MWh/y more than the other. Considering a park lifetime of 20 years and a price of energy of 0.69 €/Kwh (NPV), this translates in an increase of more than 2 M€ in revenue. This example shows the impact of mutual interference between parks, and it proves the importance of considering it in the optimization of the layout.

5.4 Conclusions

In this work we have seen how parks can benefit from mixing turbine type in their layout. The possibility of optimizing the layout considering multiple turbine types is now used inside the company, in particular for dealing with already-existing parks nearby. As we have seen in Subsection 5.3.3, this is very important to maximize the production of a site.

References

Fischetti, M. and M. Monaci (2016). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: 10.1007/s10732-015-9283-4. URL: https://doi.org/10.1007/s10732-015-9283-4.

Part III

Inter-array cable routing

Chapter 6

Optimizing wind farm cable routing considering power losses

Martina Fischetti[a] · David Pisinger[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]Management Science, Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

Publication Status: Published as: *Fischetti and Pisinger, 2017*

Reading Instructions: Main paper on the inter-array cable routing problem – OR focus

Abstract: Wind energy is the fastest growing source of renewable energy, but as wind farms are getting larger and more remotely located, installation and infrastructure costs are rising. It is estimated that the expenses for electrical infrastructures account for 15-30% of the overall initial costs, hence it is important to optimize their design. This paper focuses on offshore inter-array cable routing optimization. The routing should connect all turbines to one (or more) offshore substation(s) while respecting cable capacities, no-cross restrictions, connection-limits at the substation, and obstacles at the site. The objective is to minimize both the capital that must be spent immediately in cable and installation costs, and the future reduced revenues due to power losses. We present a Mixed-Integer Linear Programming approach to optimize the routing using both exact and math-heuristic methods. In the power losses computation, wind scenarios are handled efficiently as part of the preprocessing, resulting in a model of only slightly larger size. A library of real-life instances is introduced and made publicly available for benchmarking. Computational results on this testbed show the viability of our methods, proving that savings in the order of millions of Euro can be achieved.

6.1 Introduction

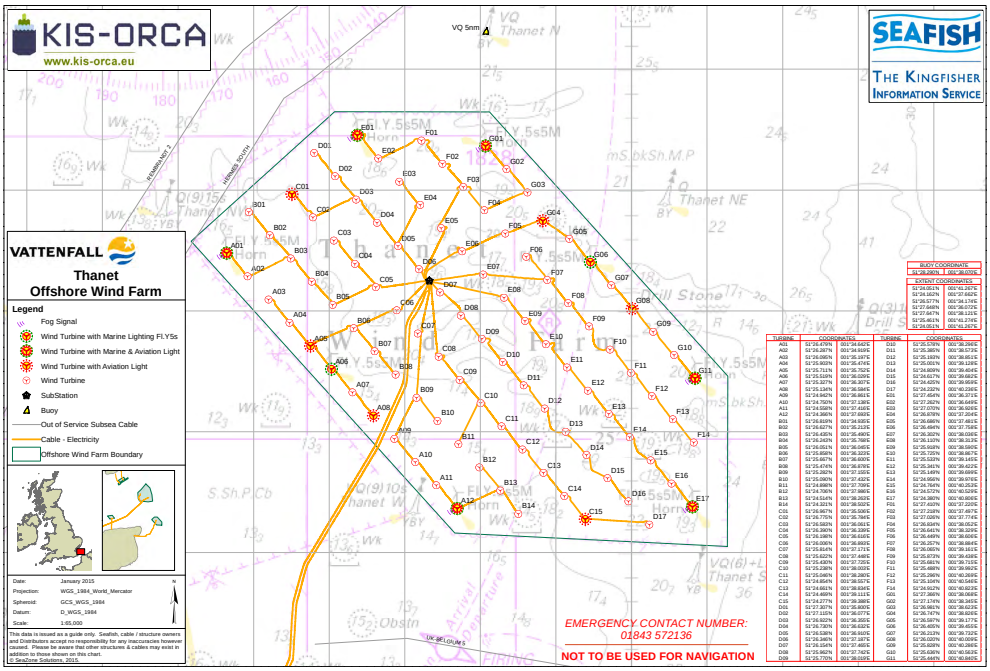
Wind power is an important technology in the transition to renewable energy, fighting climate changes. Designing a wind farm is, however, a complex process including selection of the right site, optimizing the location of each turbine (González et al., 2014; Fischetti and Monaci, 2015), establishing the infrastructure (Bauer and Lysgaard, 2015) and connecting the farm to the existing electrical grid (Qi et al., 2015). According to González et al., 2014 the expenses for electrical infrastructure account for 15-30% of the overall initial costs of an offshore wind farm. It is therefore very important to optimize the cable connections among turbines not only from an installation cost perspective, but also considering the power losses during operation.

Thanks to the collaboration with Vattenfall BA Wind it has been possible to build a detailed model including all the constraints arising in practical applications (some of which are missing in previous work from the literature) and to measure, for the first time, the savings in the long run by optimizing the layout and type of cables while taking power losses into account.

The power production of offshore turbines needs to be collected through one or more substations and then conveyed to the coast. To do that, each turbine

must be connected through a cable to another turbine, and eventually to a substation. Figure 6.1 gives an example of cable layout for a real wind park. The final cable layout, often called a *cable routing*, has a tree structure where the non-root nodes correspond to the given turbines, the substations play the role of roots, and the energy (i.e., electric current) flows from the nodes to the roots along the tree.

Figure 6.1: An example of cable routing for a real-world offshore wind park (Thanet) owned by Vattenfall—image from (Kis-orca, 2015).



A number of constraints must be taken into account when designing a feasible cable routing. First of all, the energy flow is unsplittable, i.e., the flow leaving a turbine must be supported by a single cable. In addition, each substation has a physical layout that imposes a maximum number of entering cables.

An important practical constraint is that cable crossings should be avoided. In principle, cable crossing is not impossible, but is strongly discouraged in practice as building one cable on top of another is more expensive and increases the risk of cable damages. Another restriction is due to the possible presence of obstacles in the site, e.g., nature reserves, already existing wind farms or cables, or terrain irregularities. As a consequence, the final cable routing needs

to avoid these areas.

Different types of cable with different costs, capacities and electrical resistances are available on the market. Therefore, one has to optimize also the cable type selection in order to deliver all the energy production to the substations while minimizing both direct costs (i.e., cable and installation costs) and the future revenue losses due to power losses along the cables. This latter aspect is very important in practice, in that more expensive cables/layouts can be more profitable in the long run if they limit the amount of energy lost along the cables. As far as we know, power losses were not addressed in previous work from the optimization literature, perhaps because the loss is a nonlinear function of the current flowing in the cable, hence being more difficult to handle in a Mixed Integer Linear Programming (MILP) framework. Wind park cable routing optimization has obtained considerable attention in the last years. Due to the large number of constraints and the intrinsic complexity of the problem, many studies (i.e., Dutta and Overbye, 2011a; González-Longatt and Wall, 2012; Li et al., 2008; Zhao et al., 2009) preferred to use ad-hoc heuristics. Just a few articles from the literature use Mixed Integer Linear Programming (MILP) for cable routing; see e.g. Bauer and Lysgaard, 2015; Hertz et al., 2012; Fagerfjall, 2010; Dutta, 2012; Berzan et al., 2011. To the best of our knowledge, only Cerveira et al., 2016 has considered power loss in cables. However, Cerveira et al., 2016 does not take into account variable cable loads due to fluctuating wind. Our work seems to be the only one considering obstacles and power losses over different wind scenarios. An *open vehicle routing* approach was proposed by Bauer and Lysgaard, 2015, but their model requires that only one cable can enter a turbine, a condition that is not imposed in our real-world cases. Different solution approaches were proposed in Berzan et al., 2011, where a divide-and-conquer heuristic is proposed together with an Integer Programming model that is tested on small cases involving up to 11 turbines. Hertz et al., 2012 study cable layout for onshore cases. The onshore cable routing problem has however some main differences compared to the offshore one. First of all, cables can be of two types: underground cables (connecting turbines to other turbines or to the above-ground level), and above-ground cables. In the first case, the cables need to be dug in the ground. Due to the fact that parallel lines can use the same dug hole, parallel structures are preferred (up to a fixed number). The above-ground level cables need to follow existing roads. These constraints do not exist in the offshore case. Finally, Dutta and Overbye, 2011b present a clustering heuristic for cable routing.

In this paper we present a new MILP model that is able to handle all the real-world constraints above. Our goal was the design of a practical optimization tool to be used to validate, on real cases, the potential savings resulting from

power-loss reduction.

The paper is organized as follows. Section 6.2 describes our MILP model and is divided into subsections where a basic model is first presented and then improved and extended. In particular, we show how to model power losses, and propose a precomputing strategy that is able to handle this non-linearity in a very simple way, thus avoiding sophisticated quadratic models that would make our approach impractical. We also prove that our specific version of the cable routing problem is NP-hard. Section 6.3 introduces MILP-based heuristics in the so-called *matheuristic* framework (Fischetti and Lodi, 2011; Hansen et al., 2009). We first propose a relaxation of the initial MILP model that allows one to quickly find an initial (possibly infeasible) solution, and then we introduce different ways of defining a restricted MILP to improve the current-best solution. This approach is combined with an exact solution method to obtain a hybrid heuristic/exact solution method whose performance is investigated in Section 6.4 on a testbed of real-world cases—input data being available, on request, from the first author. Section 6.5 analyzes a real case provided by Vattenfall, namely Horns Rev 3, and quantifies the savings obtained by an optimized cable routing taking power losses into account. Some conclusions and future directions of work are finally addressed in Section 6.6.

6.2 MILP model

We first introduce the basic MILP model, discuss complexity, and then describe various extensions of the model.

6.2.1 Basic model

A first step of designing a wind farm layout is to locate the turbines to maximize wind energy capture while minimizing the wake loss. Various techniques, mainly heuristic, have been proposed for this first step (Samorani, 2013; González et al., 2014; Kusiak and Song, 2010; Archer et al., 2011; Fischetti and Monaci, 2015).

Assuming that the best turbine positions have been identified, the next step is to to find an optimal cable connection among all turbines and the given substation(s), minimizing the total cable cost (excluding power losses, as these will be addressed later). Our model is based on the following requirements:

- the energy flow leaving a turbine must be supported by a single cable;
- different cables, with different capacities and costs, are available;
- the energy flow on each connection cannot exceed the capacity of the installed cable;
- a given maximum number of cables, say C, can be connected to each substation;
- cable crossing should be avoided.

Let us consider the turbine positions as the nodes of a complete and loop-free directed graph $G = (V, A)$, and all possible connections between them as directed arcs. Some nodes correspond to the substations that are considered as the roots of the distribution network, and are the only nodes that collect energy. We also add some *Steiner* nodes to add some flexibility to the cable structure, with the additional constraint that at most one cable can enter and exit each of them—hence these nodes can be left uncovered by the cable routing. As explained in Subsection 6.2.4, these dummy nodes are useful when considering obstacles in the area, or to allow for curvy connections between two nodes.

All nodes $h \in V$ have associated coordinates in the plane, that are used to compute distances between nodes as well as to determine whether two given line segments $[i, j]$ and $[h, k]$ cross each other, where $[a, b]$ denotes the line segment in the plane having nodes $a, b \in V$ as endpoints. In our application, two line segments meeting at one extreme point do not cross. Analogously, two segments do not cross if one is contained in the other, as they represent two parallel cables that can be physically built one besides the other without crossing issues.

We partition the node set V into (V_T, V_0, V_S), where V_T contains the nodes corresponding to the turbines, V_0 contains the nodes corresponding to the substation(s), and V_S contains the Steiner nodes (if any). Furthermore, let $P_h \geq 0$ denote the power production at node $h \in V$, where $P_h > 0$ for $h \in V_T$ and $P_h = 0$ for $h \in V_S$ (P_h being immaterial for $h \in V_0$).

Let T denote the set of different types of cable that can be used. Each cable type $t \in T$ has a given capacity $k_t \geq 0$ and a unit cost $u_t \geq 0$. Arc costs

$$c^t_{i,j} = u_t \cdot dist(i, j)$$

are defined for each arc $(i, j) \in A$ and for each type $t \in T$, where $dist(i, j)$ is the Euclidean distance between nodes i and j.

In our model, for each arc $(i,j) \in A$ we use a continuous variable $f_{i,j} \geq 0$ to represent the (directed) energy flow from i to j, and the binary variable $x^t_{i,j}$ with the following meaning:

$$x^t_{i,j} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is constructed with cable type } t \\ 0 & \text{otherwise.} \end{cases} \qquad (i,j) \in A,\ t \in T.$$

Finally, binary variables $y_{i,j}$ indicate whether an arc (i,j) is built with any type of cable, i.e.,

$$y_{i,j} = \sum_{t \in T} x^t_{ij}, \quad (i,j) \in A.$$

For the sake of generality, in our model we allow the costs $c^t_{i,j}$ to be defined arbitrarily. In addition, we define the undirected edge set $E = \{\{i,j\} : (i,j) \in A\}$ and generalize the non-crossing property by considering a generic input set $\mathcal{C} \subset E \times E$ of *crossing edges* with the property any two arcs (i,j) and (h,k) cannot be both constructed if $(\{i,j\},\{h,k\}) \in \mathcal{C}$.

Our basic MILP model then reads:

$$\min \sum_{(i,j) \in A} \sum_{t \in T} c^t_{i,j} x^t_{i,j} \tag{6.1}$$

$$\sum_{t \in T} x^t_{i,j} = y_{i,j}, \quad (i,j) \in A \tag{6.2}$$

$$\sum_{i \in V : i \neq h} (f_{h,i} - f_{i,h}) = P_h, \quad h \in V_T \cup V_S \tag{6.3}$$

$$\sum_{t \in T} k_t\ x^t_{i,j} \geq f_{i,j}, \quad (i,j) \in A \tag{6.4}$$

$$\sum_{j \in V : j \neq h} y_{h,j} = 1 \quad h \in V_T \tag{6.5}$$

$$\sum_{j \in V : j \neq h} y_{h,j} = 0, \quad h \in V_0 \tag{6.6}$$

$$\sum_{j \in V : j \neq h} y_{h,j} \leq 1, \quad h \in V_S \tag{6.7}$$

$$\sum_{i \in V : i \neq h} y_{i,h} \leq 1, \quad h \in V_S \tag{6.8}$$

$$\sum_{i \in V : i \neq h} y_{i,h} \leq C, \quad h \in V_0 \tag{6.9}$$

$$y_{i,j} + y_{j,i} + y_{h,k} + y_{k,h} \leq 1, \quad (\{i,j\}, \{h,k\}) \in \mathcal{C} \tag{6.10}$$

$$x^t_{i,j} \in \{0,1\}, \quad (i,j) \in A, t \in T \tag{6.11}$$

$$y_{i,j} \in \{0,1\}, \quad (i,j) \in A \tag{6.12}$$

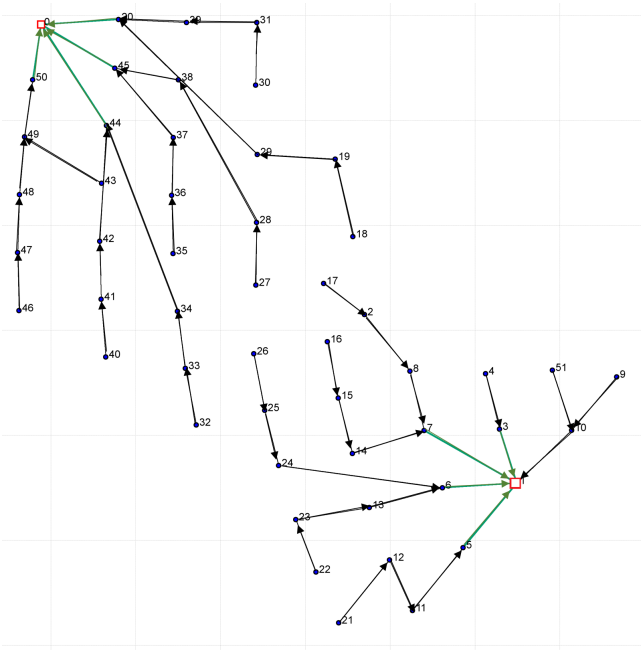
$$f_{i,j} \geq 0, \quad (i,j) \in A. \tag{6.13}$$

The objective function (6.1) minimizes the total cable layout cost. Constraints (6.2) impose that only one type of cable can be selected for each built arc, and define the $y_{i,j}$ variables. Constraints (6.3) are flow conservation constraints: the energy (flow) exiting each node h is equal to the energy entering h plus the power production of that node. Note that these constraints are not imposed for $h \in V_0$, i.e., when h corresponds to a substation. Constraints (6.4) ensure that the flow does not exceed the capacity of the installed cable. Constraints (6.5) impose that only one cable can exit a turbine and constraints (6.6) that none can exit the substation (tree structure with root in the substation). Note that the Steiner nodes can be connected or not, and if connected only one cable can enter these nodes (constraints (6.8)). In addition, (6.7) imposes that only one cable can exit a Steiner node. Constraint (6.9) imposes the maximum number of cables (C) that can enter each substation. Finally, no-cross constraints (6.10) forbid building any two incompatible arcs.

Note that the above constraints imply that the y variables define a set of connected components (one for each substation), each component containing a directed tree with out-degree at most one (i.e an anti-arborescence) rooted at a substation; see Figure 6.2 for an illustration. Circuits are not explicitly forbidden in the model, however because of (6.3) they can only arise among Steiner nodes (for which $P_h = 0$) and involve at least 3 such nodes because of (6.10). As explained in the following, this possibility is allowed in our model and used to take obstacles into account.

We finally observe that one could easily extend our model to also include a maximum number of cables entering each turbine (by adding a constraint similar to (6.9) for each turbine) or to limit the number of cables $\sum_{(i,j) \in A} x^t_{i,j}$ used for each cable type t.

Figure 6.2: A feasible cable routing; arcs are directed according to the electric current flow, i.e., towards roots (substations).



6.2.2 Problem complexity

We next address the complexity of the cable routing problem defined in the previous section. The first theorem considers the case where all turbines have the same power production $P_v = 1$ and the nodes are not associated with Euclidean coordinates, the second one considers the case where the turbines are allowed to have different production and they are associated with points on a plane.

Theorem 6.2.1. *The cable routing problem is strongly NP-hard even if* $P_v = 1$ *for all* $v \in V_T$, $|T| = 1$, $C = 2$, $|V_0| = 1$, $V_S = \emptyset$, *and* $\mathcal{C} = \emptyset$.

Proof. We prove the claim by reduction from the Weight Constrained Graph Tree Partition Problem (WGTPP). The latter problem is defined as follows. An undirected graph $G = (V, E)$ is given, with associated cost c_e for each edge $e \in E$, and weight w_v for each node $v \in V$. The problem is to partition the node set V into p disjoint clusters U_r and to build on each of them a spanning

tree T_r. The objective is to find a partition such that the overall tree cost $\sum_{r=1}^{p} \sum_{e \in T_r} c_e$ is minimized, while ensuring that each cluster satisfies a weight constraint $\sum_{v \in U_r} w_v \leq W$.

In Cordone and Maffioli, 2004 the WGTPP problem was proven to be strongly NP-hard by reduction from SAT. Actually WGTPP remains strongly NP-hard even if G is a complete graph, all nodes v have the same weight $w_v = 1$, $p = 2$, and the edge costs satisfy the triangle inequality.

Given an instance of the WGTPP on a complete graph and with $p = 2$ and $w_v = 1$ for all $v \in V$, we transform it to a cable routing problem by using the same vertices as turbines having production $P_v = w_v = 1$. An extra node is added to represent a single substation, with connection cost $-M$ with all other nodes, where M is a sufficiently large positive value. The substation can be connected to at most $C = p$ cables, and only one cable type exists ($|T| = 1$) with capacity $k_1 = W$. No Steiner nodes are present ($V_S = \emptyset$), and $\mathcal{C} = \emptyset$. By construction, an optimal solution of the cable routing problem yields the required optimal WGTPP partition.

□

Interestingly for our wind-farm application, our cable routing problem remains NP-hard even in its *2D-Euclidean version*, i.e., when nodes have associated coordinates in the plane, costs depend on the Euclidean distance, and set $\mathcal{C}$ is defined according to the geometrical crossing property between line segments in the plane.

Theorem 6.2.2. *The 2D-Euclidean cable routing problem is NP-hard even when $V_S = \emptyset$, $C = \infty$, and $|T| = 1$.*

Proof. We use a reduction from the following well-known NP-complete problem Garey and Johnson, 1979:

> `PARTITION`: Given a set of n positive integers $r_1, \cdots, r_n$, does there exist a set $Q \subset \{1, \cdots, n\}$ such that $\sum_{h \in Q} r_h = \sum_{h=1}^{n} r_h/2$?

Given any instance of `PARTITION`, we define an instance of our cable routing problem as follows. We define $V_0 = \{0\}$, $V_T = \{1, \cdots, n\}$, $V_S = \emptyset$, and $P_h = r_h$ for $h = 1, \cdots, n$. We define only one type of cable (i.e., $T = \{1\}$) with unit cost $u_1 = 1$ and capacity $k_1 = \sum_{h=1}^{n} r_h/2$, and set $C = \infty$. As to point coordinates

in the plane, we position the substation node 0 at coordinates $(0, 0)$, while all turbine points $h \in V_T$ are located at coordinates $(1, 0)$. As all points lay on a line, no crossing can arise as the line segments corresponding to the arcs of G are parallel. Furthermore note that, as cable cost depends on the cable length (which is zero between overlapping points), it is always more convenient to connect turbines one to each other than to the substation (until the cable capacity is exceeded). By construction, the answer to `PARTITION` is yes if and only if the optimal cable routing problem has an optimal value of 2, meaning that only 2 cables are connected to the substation, each with saturated capacity. This completes the proof of correctness of our reduction.

□

6.2.3 Improved no-cross constraints

No-cross constraints (6.10) in our basic model have two main drawbacks: they are weak in polyhedral terms, and their number can be very large (up to $O(|V|^4)$ constraints). We therefore propose a clique strengthening of these constraints, that exploits constraints (6.5)-(6.7) to reduce their number and to typically improve their quality. For any node triple (a, b, k), let the *clique* arc subset $\mathcal{Q}(a, b, k)$ be defined

$$\mathcal{Q}(a, b, k) = \{(a, b), (b, a)\} \cup \{(k, h) \in A : (\{a, b\}, \{k, h\}) \in \mathcal{C}\} \tag{6.14}$$

Theorem 6.2.3. *The following* improved no-cross constraints *are valid for model* (6.1)-(6.13)*:*

$$\sum_{(i,j)\in\mathcal{Q}(a,b,k)} y_{i,j} \leq 1, \qquad a, b, k \in V,\ |\{a, b, k\}| = 3 \tag{6.15}$$

Proof. Proof. Let (y, x, f) be any feasible solution of model (6.1)-(6.13). As the solution cannot contain 2-node circuits, we have $y_{a,b} + y_{b,a} \leq 1$. If $y_{a,b} = y_{b,a} = 0$, the claim follows from the out-degree inequalities (6.5)–(6.7) that imply $\sum_{h\in V:h\neq k} y_{k,h} \leq 1$. Otherwise, assume without loss of generality $y_{a,b} = 1$, and observe that the no-cross constraint (6.10) forbids the selection of any arc (k, h) such that $(\{a, b\}, \{k, h\}) \in \mathcal{C}$, hence $y_{k,h} = 0$ for all $(k, h) \in \mathcal{Q} \setminus \{(\dashv, \lfloor), (\lfloor, \dashv)\}$. This proves that y satisfies all inequalities (6.15), as claimed. □

Note that the improved no-cross constraints (6.15) can replace (6.10) in model (6.1)-(6.13), thus reducing their number from $O(|V|^4)$ to $O(|V|^3)$. However,

although typically better, constraints (6.15) do not dominate (6.10) in the sense that a *fractional* solution y^* can violate a certain inequality (6.10) but no improved inequality (6.15). To see this, it is enough to consider a fractional solution y^* with $y^*_{a,b} = y^*_{b,a} = y^*_{h,k} = y^*_{k,h} = 1/3$, where $(\{a,b\},\{h,k\}) \in \mathcal{C}$ so (6.10) is violated, whereas $Q(a,b,k) = \{(a,b),(b,a),(k,h)\}$, and thus (6.15) is not violated.

6.2.4 Modeling obstacles and curvy cable connections

In practical applications, some obstacles can be present in the site, meaning that some areas of the site cannot be crossed by cables. As an illustration, consider the real-world offshore wind park of Figure 6.3 where two big obstacles are present—they are represented by the two polygons in the figure. All the given 29 turbines must be connected to the offshore substation, represented by the square node 0 in the figure. The substation has a limit of $C = 8$ connected cables. In this example we considered three types of cables:

- type 1 can support up to 5 turbines and has a price of 135 Euro/m;
- type 2 can support up to 7 turbines and has a price of 250 Euro/m;
- type 3 can support up to 9 turbines and has a price of 370 Euro/m.

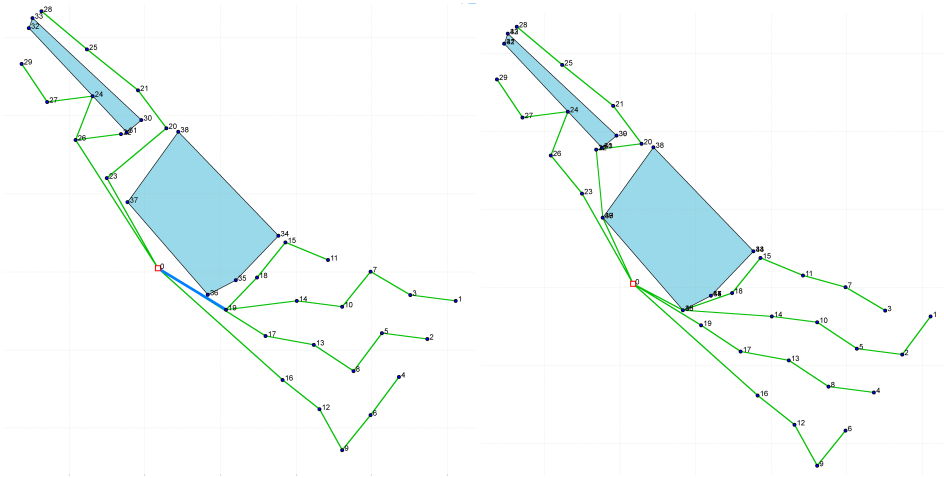
The prices above are publicly available and do not include installation costs. In Figure 6.3, type 1 cables are depicted in green, while type 2 cables are in bold blue (type 3 is not selected in any solution).

To define each obstacle area, viewed as a closed polygon, in our model we introduce artificial Steiner nodes (with no production) at the vertices of polygon and connect them through a circuit made by a artificial zero-cost cable that is forced in the solution by fixing the corresponding binary variable to 1. In this way, the final cost of the layout is not affected by the artificial cables but, due to the no-cross constraints, the actual cables cannot cross the obstacles. Referring to Figure 6.3 (left subfigure), only 9 Steiner nodes were introduced, namely:

- nodes 30, 31, 32, and 33, used to define the obstacle in the top-left corner.
- nodes 34, 35, 36, 37, and 38, used to define the big central obstacle;

Figure 6.3 (left) shows the optimized cable routing when considering only the 9 Steiner points above.

Figure 6.3: A real-world offshore wind farm with obstacles modeled through Steiner nodes. The layout on the right uses additional Steiner nodes to deal with curvy cable connections, and allows for a 7% cost saving.



Looking at the final layout, it is clear that the cable routing is rather expensive due to the non-flexibility of the cables (note, indeed, that our model can only deal with straight-line cable connections). For example, turbine 18 cannot be connected directly to the substation, because the straight edge $[18, 0]$ would cross the obstacle. In this particular case, this has a significant impact on the final cost, as 18 is connected to 19 and therefore arc $(19, 0)$ needs a more expensive cable, since it has nine turbines connected; see the bold blue line in Figure 6.3 (left).

In real world, however, cables can be curvy so this connection would be possible. To simulate the curvy form of some cable connections, we can again use Steiner nodes. Our idea is to put some of them around the obstacles, so that the final solution can possibly use them to circumvent the obstacles. In particular, in the above example, one can put a new Steiner node over each of the artificial nodes delimiting the obstacles, plus two extra Steiner nodes overlying the two obstacle-points closer to the substation (which are in the most critical area). In this way a curvy cable can connect 14 to the substation and another can connect 18 to the substation. These connections support less flow (from five turbines each) and therefore can use a cheaper cable. Note, that two overlapping cables are not considered as a cable crossing since they can be laid in parallel. More

specifically, in Figure 6.3 (right) the additional Steiner points used to add flexibility are:

- nodes 39, 40, 41, and 42 overlying respectively nodes 30, 31, 32, and 33;
- nodes 43, 44, 45, and 46 overlying respectively nodes 34, 35, 36, and 37;
- nodes 47, 48, and 49 overlying respectively nodes 35, 36 and 37.

Figure 6.3 (right) shows the new optimized cable routing, which uses only the less-expensive cable type 1 and allows for a cost saving of about 7% with respect to the previous one.

6.2.5 Modeling cable losses

In this section we extend the previous model to take cable losses into account. We first explain the physics behind the problem and next we introduce an alternative way to handle power losses through a simple preprocessing to be applied to the input data of our MILP model.

Physically, power losses are proportional to the square of the current. If we indicate with $g^t_{i,j} \geq 0$ the current actually passing through the cable of type t on arc (i, j), the total power loss in all the cable layout can be computed as

$$\sum_{(i,j)\in A} \sum_{t\in T} Q^t_{i,j} \, (g^t_{i,j})^2 \tag{6.16}$$

where $Q^t_{i,j}$ is a positive constant depending on the cable type and length that we define as

$$Q^t_{i,j} = R^t \cdot dist(i, j) \tag{6.17}$$

and R^t is the electrical resistance of cable type t, in Ω/m. If we want to estimate the value of these losses (in order to compare them with the layout price) we need to multiply the result by K_{euro}. K_{euro} is the cost for each MW of production.

Of course the current passing through each cable depends on the production of the connected turbines and is limited by the capacity of the used cable. Therefore, the new variables $g^t_{i,j}$ need to be linked to the remaining variables in the model through appropriate constraints.

As the electrical currents depend on the wind scenarios, one could think of a 2-stage Stochastic Programming model where x and y act as first stage variables, and we have a copy of variables $g^t_{i,j}$ for each wind scenario. This would lead to a really huge non-linear model, that would be very difficult to solve even for small instances.

Having understood the physics behind cable losses, we prefer to stick to a much simpler (and practically very effective) approach to deal with cable losses implicitly according to the following idea. We consider the MILP model without cable losses on a modified instance where each cable type is replaced by a series of *sub-cables* with discretized capacity and modified cable cost taking both installation costs and revenue losses into account.

As an illustration, consider a common situation where all turbines in the wind farm are identical, hence the maximum power production P_h of each turbine can be assumed to be 1. This means that we can express cable capacities as the maximum number of turbines supported by each cable type. Now consider a certain cable type t that can support up to k_t turbines. We replace it by k_t *sub-cable* types of capacity $f = 1, \ldots, k_t$ whose actual unit cost is computed by adding the cable/installation unit cost (u_t) and the unit power-loss cost (say $loss_{t,f}$) computed by considering the current produced by exactly f turbines. As unit costs increase with f, the optimal solution will always select the sub-cable type f supporting exactly the number of turbines connected, hence the approach models power losses in a correct way.

Note that the final sub-cable list would in principle contain $\sum_{t \in T} k_t$ different sub-cable types. However, for each sub-cable capacity $f = 1, \ldots, k_{MAX} = \max\{k_t : t \in T\}$ one only needs to keep the sub-cable with capacity f and minimum unit cost, i.e., only k_{MAX} "undominated" sub-cable types need to be considered in practice. This figure is not too large in practical cases, and can effectively be handled by our solution algorithms.

The above approach also allows us to consider multiple wind scenarios (and hence different current productions of the turbines) without the explicit need of second-stage variables, thus keeping the model size manageable. Indeed, one can just precompute the sub-cable unit costs by considering a weighted average of the loss cost under different wind scenarios. In our computational study, the wind energy experts computed the loss contribution to the unit cost for a 3-phase sub-cable of capacity f as

$$loss_{t,f} = 3 \sum_{s \in S} \pi_s (f I^s)^2 R^t K_{euro} \tag{6.18}$$

where S is the set of wind scenarios under consideration, π_s is the probability of scenario $s \in S$, and I^s is the current produced by a single turbine under

wind scenario s assuming negligible wake effects, i.e., all turbines produce the same electric current.

Example: We will next show an example of how the power-loss prices are computed for a cable set named cb05 in Section 6.3.3 (these are realistic cables, though they do not refer to any specific cables on the market). We will consider the wind statistics from a real-world wind park in Denmark, namely Hors Rev 1 (named wf01 in Section 6.3.3).

Without cable losses, the cable cost would be taken directly from the cable information provided by the company, and would correspond to the sum of cable and installation costs—as reported in the last column of Table 6.1. When cable losses come into play, we need to modify the cable set and its prices, according to the strategy presented above. As cable type 1 supports up to 10 turbines, we need 10 sub-cables to deal with it, while 4 sub-cables are enough for cable type 2; see Table 6.2.

As to sub-cable prices, they need to consider also the power losses incurred under different scenarios. We used formula (6.18) with π_s and I^s taken from the actual wind statistics from the specific site. Parameter $K_{euro} = 0.68$ was computed by Vattenfall's experts by considering a cable lifetime of 25 years, a WACC of 8%, a warranted price of 0.10 €/KWh for 10 years and then a market price of 0.02 €/KWh, while resistance R_t is defined according to Table 6.1.

The prices considering cable cost, installation and losses for cb05 in the Horns Rev 1 case are shown in Table 6.2. Notice that, for each cable type, the sub-cable costs are monotonically increasing with the number of turbines supported, therefore in an optimal layout any sub-cable will support a number of turbines exactly equal to its maximum capacity.

For example, in row number 4 of Table 6.2 we consider the situation where cable type 1 is used to support the electrical current of 4 turbines. Its unit price is computed as 440 €/m for immediate costs, plus about 8.87 €/m for the estimated power losses related to the electrical currents produced by 4 turbines under the given wind scenarios. Note that the additional cost is non-linear, e.g., it is equal to about 33.54 €/m for 8 turbines. However, this non-linearity is handled in the input precomputation, without affecting the linearity of the underlying MILP model.

Table 6.1: Cable information for cb05.

cables	type	n. of 2MW turb. connected	resistance [Ohm/km]	cable price [€/m]	install. price [€/m]	total price [€/m]
cb05	1	10	0.13	180	260	440
	2	14	0.04	360	260	620

Table 6.2: Cable prices precomputed considering fixed costs and power losses for wf01 with cb05.

cable type	n. of 2MW turb. supported	price [€/m]
1	1	441.16
	2	442.71
	3	445.27
	4	448.87
	5	453.50
	6	459.15
	7	465.83
	8	473.54
	9	482.28
	10	492.04
2	11	639.77
	12	643.41
	13	647.36
	14	651.63

The comparison between Table 6.1 and 6.2 shows the impact of considering losses on cable prices. While from a installation perspective the cost for each cable type is fixed, now it varies depending on how many turbines are connected. As we will see, this can have a large impact on the optimal cable routing.

6.2.6 No-cross constraint separation

As the number of no-cross constraints (6.15) on the complete graph $G = (V, A)$ can be very large for real-world instances, we decided not to include them in

the model that is passed to the MILP solver. Instead, we generate them on the fly, during the MILP solver execution. To this end, we implemented a cut separation "callback" function that is automatically invoked by the solver to verify whether the current solution satisfies (6.15). This callback function is invoked both for (possibly fractional) solutions arising when solving the LP relaxation at any given branching-tree node, and for integer solutions generated by the internal MILP-solver heuristics.

Our separation function receives the (possibly fractional) solution y^* on input, and scans all node triples (a, b, k) with $a < b$ to check whether the corresponding improved no-cross constraint (6.14) is violated by y^*. Pairs (a, b) with $y^*_{a,b} = y^*_{b,a} = 0$ are skipped as they cannot lead to a violation constraint as y^* satisfies the out-degree inequalities (6.5)–(6.7).

Violated constraints, if any, are returned to the MILP solver in an appropriate format, so they can automatically be added to the current model. If no violated constraint (6.14) is found and y^* is fractional, we also apply a similar separation procedure to possibly generate violated constraints (6.14)(though this occurs in very rare cases).

6.3 Solution method

Since the MILP solver cannot solve large problems to optimality and often fails in even finding a feasible solution, we propose a matheuristic to find high-quality solutions in reasonable time.

6.3.1 A relaxed model

To be able to find meaningful (through possibly infeasible) initial solutions in very short computing times, we relax our MILP model by allowing it to produce disconnected solutions. To this end, for each $h \in V_T$ we replace the corresponding equality in (6.5) by a less-or-equal inequality. This however would not affect the final solutions due to the presence of the flow equilibrium equations (6.3). So we also relax the latter by allowing for a current loss in some nodes. This is obtained by introducing a slack continuous variable $l_h \geq 0$ for each $h \in V$, that indicates the energy lost at node h, and by replacing (6.3)

by

$$\sum_{i \in V : i \neq h} (f_{h,i} - f_{i,h}) + l_h = P_h, \ h \in V_T \cup V_S \tag{6.19}$$

$$0 \leq l_h \leq k_{MAX}, \ h \in V_T \cup V_S \tag{6.20}$$

where $k_{MAX} = \max\{k_t : t \in T\}$ is the maximum cable capacity. The objective function coefficient of the new variables l_h is set to a very large positive constant, say $M \gg 0$, thus ensuring that a connected solution would always have a lower cost compared with a disconnected one.

To avoid solutions where no arcs are built, we add to the model a constraint that requires to install a total capacity in the arcs entering the substations that is sufficient to support the total turbine production, namely

$$\sum_{t \in T} \sum_{(i,j) \in A : j \in V_0} k^t x^t_{i,j} \geq \sum_{h \in V_T} P_h. \tag{6.21}$$

Our computational experience showed that a black-box MILP solver applied to the relaxed model above is typically able to find, in a few seconds, a feasible (possibly disconnected) first solution and to quickly proceed in the tree enumeration to discover better and better ones.

6.3.2 Matheuristics

As its name suggests, a *matheuristic* (Fischetti and Lodi, 2011; Hansen et al., 2009) is the hybridization of mathematical programming with metaheuristics. The hallmark of this approach is the possibility of designing sound heuristics on top of a black-box MILP solver, by just changing its input data in a way that favors finding a sequence of improved solutions. In our settings, the black-box MILP solver is our exact method described in the previous section, where no-crossing constraints are separated on the fly as described in the next Section 6.2.6.

In our setting, the matheuristic is used as a refining tool that receives a certain solution H (described by its associated vector, say (x^H, y^H)) and tries to improve it by solving a restricted MILP "tailored around H" by fixing $y_{a,b} = 1$ for a suitable-defined subset of the y variables with $y^H_{a,b} = 1$. Note that this variable-fixing scheme is very powerful in our context as every time a certain $y_{a,b} = 1$ is fixed on input, one can forbid all possible crossing arcs, in a preprocessing phase, by just setting $y_{i,j} = 0$ for $(i,j) \in \mathcal{Q}(\dashv, \lfloor, \|) \setminus \{(\dashv, \lfloor)\}$ for all

$k \in V \setminus \{a, b\}$. This is very important for the success of our heuristic, as the restricted problem becomes much easier to solve due to the large number of variables fixed to 0 or to 1, and because of the fact that many relevant no-cross constraints are implicitly imposed by preprocessing.

In our implementation, we iteratively apply our refining matheuristic to the current best solution H available. At each iteration, we temporarily fix to 1 some y variables according to a certain criterion (to be described later), and apply the preprocessing described above to temporarily fix some other y variables to zero. We then apply the MILP solver to the corresponding restricted problem, and we warm start the solver by providing the current solution (x^H, y^H). We abort the execution as soon as a better solution is found, or a short time limit of a few seconds is reached. Then all fixed variables are unfixed, and the overall approach is repeated until a certain overall time limit (or maximum number of trials) is reached.

The very first solution H passed to the matheuristic plays an important but unpredictable role in determining the quality of the final solution. As a matter of fact, starting with a very bad (disconnected) solution can sometimes lead to very good final solutions. This behaviour suggests a *multi-start* strategy where a number of different initial solutions are generated and then iteratively improved by using our overall matheuristic. In our implementation, the initial solutions are defined by taking the very first (typically highly disconnected) solution found by the MILP solver when applied to our relaxed model with a random objective function where all the y variables have a random cost. (Note that the cost of the slack variables l_h introduced in (6.19) remains unchanged, meaning that disconnected solutions are still penalized.) To enhance diversification even further, we also use different input values for the MILP-solver's random seed parameter.

We next describe four possible variable-fixing criteria.

Our first criterion, called RANDOM, follows a simple *random variable-fixing* scheme that fixes variables $y_{a,b}$ with $y^H_{a,b} = 1$ with a certain probability, e.g., 50%.

Our second criterion, DISTANCE, uses a problem-specific strategy to choose the arc-fixing probability. To be more specific, the arc-fixing probability is related with the distance to the substation, namely: the arcs closer to the substation(s) are fixed with a larger/smaller probability.

Our third criterion, SWEEP, is specific for wind farms with only one substation.

In these cases, the optimal solutions tend to divide the wind farm into *radial sectors* emanating from the substation. To take advantage of this property, we tentatively partition the wind farm into sectors, and we then iteratively reoptimize each sector by fixing all the arcs that involve two nodes not belonging to it.

To be more specific, within SWEEP all nodes are initially ordered according to their angle with the substation. This produces a cyclic node sequence where the last-ranked node is followed by the first-ranked one. A *seed* node is randomly selected and a *sector* is defined by picking the $SPAN$, say, nodes that follow it in the cyclic ordered sequence. (In our implementation, we set $SPAN$ as a certain percentage of the total number of turbines.) The y variables associated with the arcs connecting two nodes outside this sector are fixed to their value in y^*, while any other y variable is left unfixed and the subproblem is reoptimized though our black-box MILP solver. If an improved solution is found, we select the seed s for the new iteration to be close to the previous one (i.e., in the cyclic sequence we randomly pick a node in the interval [s-2, s+2]). If the solution is not improved, instead, the new seed is taken by moving of 5 (say) nodes forward in the sequence. The SWEEP heuristic ends when the seed s moved along a complete cycle without improving the current solution.

Our fourth criterion, STRINGS, is similar to SWEEP but at each iteration it defines the sector to be reoptimized as follows. For each arc (i, r) with $y^*_{i,r} = 1$ that enters the substation, say r, we define the node set S_i containing all the nodes that reach the substation r passing through node i. In other words, S_i contains all the predecessors of node i in the anti-arborescence corresponding to y^*. Node sets S_i's are then sorted according to the angle formed by segment $[i, r]$ with respect to a vertical line passing through r. At each iteration, we choose a seed set S_{seed} and define the current *sector* to be optimized by taking $S_{seed} \cup S_{seed+1} \cup \cdots \cup S_{seed+SPAN}$ (where $SPAN$ is a given parameter and subscripts are taken in a cyclic way).

6.3.3 Computational tuning

To test how our four matheuristic algorithms perform on realistic data, we created a dataset of 11 synthetic wind farms. All instances are difficult as they involve a large number of turbines (from 60 to 93) and in some cases one or more obstacles.

Our matheuristics have been implemented on top of the state-of-the art commercial MILP solver IBM ILOG CPLEX 12.6. We used 3 instances for the

parameter tuning of each algorithm (training set), and the remaining 8 to compare the matheuristics (test set). Each test has been run four times with four different random seeds, thus producing different final solutions. All matheuristics have been run with a time limit of 15 minutes on an Intel Xeon CPU X5550 running at 2.67GHz.

For the sake of space, we do not report here the detailed results of our experiments. Instead, we next summarize our findings.

- Matheuristic RANDOM is quite effective in its first iterations as it quickly finds good feasible solutions, though its performance becomes less satisfactory when the current solution becomes almost optimal.
- Matheuristic DISTANCE does not provide significantly better results than RANDOM, both when the fixing probability is larger for the arcs closer to the substation(s) and vice-versa. So RANDOM should be preferred as it is simpler and less prone to overtuning.
- Matheuristics SWEEP and STRINGS have a comparable performance and each of them tends to outperform RANDOM when the current solution is close to optimality
- The best overall results are obtained by using the above matheuristics in a combined way, as described in Section 6.3.4.

6.3.4 The overall algorithm

Our final algorithm is a mixture of matheuristic and exact solvers. To be specific, we first apply the RANDOM matheuristic with 50% fixing probability for 30 iterations, so as to quickly obtain a good solution. Then we apply STRINGS with $SPAN = 3$, and finally SWEEP with $SPAN = 0.3 \cdot |V_T|$. This matheuristic sequence is applied 5 times, in a multi-start vein, each time restarting from scratch from a different random initial solution. At each restart, the initial solution is defined as the very first (typically disconnected) solution found by the exact MILP solver (CPLEX 12.6) when applied to the relaxed model of Subsection 6.3.1 with a random objective function.

After the 5th restart, the exact MILP solver is applied to the original model without any heuristic variable fixing, using the best-available solution to warm-start the solver. In other words, our overall approach is intended to reach a proven optimal solution (if time limit permits). To this end, we first obtain

very good heuristic solutions, in the matheuristic phase, and then we switch to the exact method to obtain a valid lower bound—and eventually to converge to proven optimality.

6.4 Computational analysis

This section is aimed at testing the performance of the MILP-based approach described in Subsection 6.3.4, using real-world data.

6.4.1 Test instances

For benchmark purposes we collected the data of five different real wind farms in operation in United Kingdom and Denmark. This data is available, on request, from the first author.

Table 6.3 summarizes the relevant information of the different sites for these five instances, named wf01, wf02, wf03, wf04 and wf05 in what follows. Figure 6.4 shows the different wind farms locations.

Figure 6.4: The real-world wind farms used in our tests.

We estimated the maximum number of connections to the substation, namely the input parameter C, by looking at the existing cable layout publicly available at (Kis-orca, 2015) and (4cOffshore, 2015). The corresponding constraint also determines the cable types to be used, which are reported in the last column of Table 6.3.

Our first wind farm (wf01) refers to Horns Rev 1, one of the oldest large-scale wind parks in the world. It was built in 2002 in the North Sea, about 15 km from the Danish shore, and produces around 160 MW. Our second wind farm (wf02) is Ketish Flats. Ketish Flats is located close to Kent in South East England, and can produce up to 140MW. It is a near-shore wind farm, so it is connected to the onshore electrical grid without any offshore substation. Nevertheless, only one export cable is connected to the shore, therefore we took the starting point of the export cable to act as a substation. This is handled by setting $C = \infty$ as there is no physical substation limitation in this case.

Our third wind farm (wf03) is Ormonde located in UK as well, but in the Irish Sea. It has a total capacity of 150 MW. Close to Kentish Flats (wf02) there is also Thanet (wf05), a bigger wind park with capacity 300 MW. When it was opened, in 2010, Thanet was the biggest offshore wind farm in the world. Finally wf04 refers to the DanTysk offshore wind farm, located west of the island of Sylt and directly on the German-Danish border. With a total of 80 wind turbines (288MW), DanTysk can provide up to 400,000 homes with green energy. All the considered sites are owned by Vattenfall.

The turbines in each wind park are of the same type, so we can assume $P_h = 1$ for all turbine nodes and we can express the cable capacities as the maximum number of turbines that it can support. Table 6.4 reports the capacity for the different cable types. We considered 5 different sets of real cables, named cb01, cb02, cb03, cb04, and cb05. As we know their capacity, resistance and price, we could precompute their unit costs both with and without power losses, following the strategy we proposed in Subsection 6.2.5.

We already observed that power losses depend on the current flowing in the cables, that in turn depends on the average wind conditions within the site. We computed the cable-loss prices as a combination of real measured data and estimations based on Weibull distributions.

Each combination of site (i.e. wind farm) and feasible cable set represents an

Table 6.3: Basic information on the real-world wind farms we used for tests.

name	site	turbine type	n. of turbines	C	allowed cables
wf01	Horns Rev 1	Vestas 80-2MW	80	10	cb01-cb02-cb05
wf02	Kentish Flats	Vestas 90-3MW	30	∞	cb01-cb02-cb03-cb04-cb05
wf03	Ormonde	Senvion 5MW	30	4	cb03-cb04
wf04	DanTysk	Siemens 3.6MW	80	10	cb01-cb03-cb04-cb05
wf05	Thanet	Vestas 90-3MW	100	10	cb04-cb05

Table 6.4: Basic information on the real-world cables we used for tests.

		n. of turbines supported			
cables	type	2MW	3MW	5MW	3.6MW
cb01	1	7	5	3	4
	2	11	8	4	6
	3	13	9	6	8
cb02	1	7	5	3	5
	2	12	8	5	7
cb03	1	12	8	5	7
	2	23	16	5	14
cb04	1	9	7	4	6
	2	21	15	9	13
cb05	1	10	7	4	6
	2	14	10	6	8

instance in our testbed, resulting in a total of 29 instances. Table 6.5 reports the main characteristics of our testbed. Each instance is identified by a number from 1 to 29 (first column in the table), and corresponds to an existing wind farm layout (second column) and to a set of possible cables (third column). Some cable sets refer to immediate costs of cables (i.e., CAPital EXpenditure, indicated as "capex" in the table) while others to costs including losses.

6.4.2 Tests

We first test the MILP model we developed in Section 6.2. To understand the capability of the model alone, we solve our real-world instances with a 10 hours time limit (Intel Xeon CPU X5550 running at 2.67GHz, CPLEX 12.6). The results are shown in Table 6.6. The first column indicates the

Table 6.5: Our testbed.

number	wind farm	cable set
01	wf01	wf01_cb01_capex
02		wf01_cb01
03		wf01_cb02_capex
04		wf01_cb02
05		wf01_cb05_capex
06		wf01_cb05
07	wf02	wf02_cb01_capex
08		wf02_cb01
09		wf02_cb02_capex
10		wf02_cb02
11		wf02_cb03
12		wf02_cb04_capex
13		wf02_cb04
14		wf02_cb05_capex
15		wf02_cb05
16	wf03	wf03_cb03_capex
17		wf03_cb03
18		wf03_cb04_capex
19		wf03_cb04
20	wf04	wf04_cb01_capex
21		wf04_cb01
22		wf04_cb03
23		wf04_cb04
24		wf04_cb05_capex
25		wf04_cb05
26	wf05	wf05_cb04_capex
27		wf05_cb04
28		wf05_cb05_capex
29		wf05_cb05

instance considered (refer to Table 6.5 for details), the second the best solution found within the time limit, the third one reports the quality of the solution as computed by the solver and the last column reports the time used to find the proven optimal solution (36000.00 if the time limit is exceeded without finding it). It is seen that the MILP model alone works well for smaller instances: for instances 07 to 19 the optimal solution is found in a few seconds. However,

these instances refer to the smallest wind farms in our data set (Ormonde and Kantish Flats). When considering bigger wind parks (as DanTysk and Thanet) the MILP solver alone is not able to find even a feasible solution within 10 hours. Modern wind parks are often of the size of Thanet and DanTysk, so the algorithm should be designed to solve such instances.

Table 6.6: Solution quality obtained using our MILP model alone with a time limit of 36000 secs (– means infeasible solution).

Inst.	bestsol	% error LB	time
01	19437282.55	0.079	36000.00
02	–	–	36000.00
03	22611988.67	0.007	36000.00
04	24446239.21	0.470	36000.00
05	23482483.25	0.327	36000.00
06	–	–	36000.00
07	8555171.40	0.000	2.78
08	8806838.99	0.000	5.03
09	10056670.31	0.000	1.67
10	10303320.51	0.000	5.34
11	9200184.65	0.000	20.75
12	8604208.93	0.000	0.97
13	8933494.59	0.000	5.67
14	10173931.59	0.000	1.12
15	10348430.63	0.000	12.34
16	8054844.90	0.000	25.11
17	8560008.68	0.000	145.91
18	8357195.91	0.000	117.08
19	9178499.88	0.000	10029.88
20	–	–	36000.00
21	–	–	36000.00
22	–	–	36000.00
23	44421681.46	2.490	36000.00
24	–	–	36000.00
25	–	–	36000.00
26	22336016.56	3.330	36000.00
27	–	–	36000.00
28	–	–	36000.00
29	–	–	36000.00

Therefore, we now evaluate the capability of the matheuristic algorithm we presented in Subsection 6.3.4 to solve these real-world cable routing instances. To this end, we performed different runs with a time limit of 60, 300, 600, 1800, 3600 (1h), 36000 (10h), and 86400 secs (24h) on our computer.

Table 6.7 reports the quality of the solution found compared with the best solution known. For any instance, LB denotes the best-known lower bound computed by the exact solver after 24 hours of computing time. The table

reports the instance number, the best-known feasible solution (best) and the associated percentage error with respect to LB (%error wrt best/LB), the computing time needed to get the best solution (time, in CPU secs), and then the percentage error with respect to LB and best at the various time limits. Solutions with a relative error of at most 0.01% satisfy the default optimality tolerance of our solver (10^{-4}), so they are considered optimal. Entries "–" refer to an infeasible (i.e., disconnected) solution found by the matheuristic in its early stage.

Table 6.7: Solution quality obtained using our matheuristic with different time limits. (– means infeasible solution).

				60 s.		300 s.		600 s.		1800 s.		3600 s.		10 h		24 h	
		% error	time	%error wrt		%error wrt		%error wrt		%error wrt		%error wrt		%error wrt		%error wrt	
Inst.	best sol.	best/LB	to best	LB	best	LB	best	LB	best	LB	best	LB	best	LB	best	LB	best
01	19436700.18	0.01	14748.8	1.75	1.74	0.08	0.07	0.08	0.07	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
02	21403410.11	0.09	86400.5	–	–	0.29	0.19	0.09	0.00	0.09	0.00	0.09	0.00	0.09	0.00	0.09	0.00
03	22611988.67	0.01	4621.3	0.39	0.38	0.18	0.17	0.03	0.02	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
04	24445688.02	0.25	86400.9	–	–	0.45	0.21	0.27	0.02	0.25	0.00	0.25	0.00	0.25	0.00	0.25	0.00
05	23482483.25	0.17	86400.3	2.22	2.04	0.64	0.46	0.58	0.41	0.58	0.41	0.24	0.07	0.24	0.07	0.17	0.00
06	24768927.72	0.97	86401.3	2.85	1.86	1.01	0.04	0.97	0.00	0.97	0.00	0.97	0.00	0.97	0.00	0.97	0.00
07	8555171.40	0.00	43.0	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
08	8806838.99	0.00	58.4	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
09	10056670.31	0.00	28.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10303320.51	0.00	59.4	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
11	9200184.65	0.00	174.4	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
12	8604208.93	0.00	23.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	8933494.59	0.00	73.2	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
14	10173931.59	0.00	24.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10348430.63	0.00	88.0	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
16	8054844.90	0.00	247.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	8560008.68	0.00	1278.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	8357195.91	0.00	354.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	9178499.88	0.01	1718.1	0.93	0.92	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
20	38977593.84	5.10	86400.4	–	–	6.62	1.45	5.57	0.45	5.10	0.00	5.10	0.00	5.10	0.00	5.10	0.00
21	44857986.73	4.53	86401.3	–	–	4.88	0.34	4.53	0.00	4.53	0.00	4.53	0.00	4.53	0.00	4.53	0.00
22	40949573.29	2.11	86400.1	2.71	0.59	2.17	0.06	2.11	0.00	2.11	0.00	2.11	0.00	2.11	0.00	2.11	0.00
23	44421681.46	2.54	86400.1	4.89	2.29	2.54	0.00	2.54	0.00	2.54	0.00	2.54	0.00	2.54	0.00	2.54	0.00
24	50379247.34	7.94	86400.5	12.12	3.88	8.15	0.20	7.94	0.00	7.94	0.00	7.94	0.00	7.94	0.00	7.94	0.00
25	52331587.72	4.64	86400.9	–	–	9.13	4.29	7.43	2.67	6.59	1.87	4.71	0.06	4.64	0.00	4.64	0.00
26	22337935.84	3.49	86400.6	3.98	0.48	3.49	0.00	3.49	0.00	3.49	0.00	3.49	0.00	3.49	0.00	3.49	0.00
27	23362025.61	3.42	86400.2	5.52	2.03	3.85	0.42	3.45	0.03	3.42	0.00	3.42	0.00	3.42	0.00	3.42	0.00
28	26637602.25	2.57	86400.5	–	–	–	–	6.20	3.54	3.32	0.73	3.32	0.73	3.32	0.73	2.57	0.00
29	27295289.87	3.87	86401.7	–	–	–	–	–	–	4.84	0.94	3.87	0.00	4.09	0.21	4.09	0.22

Table 6.7 reports the quality of the solution found compared with the best solution known. More specifically, the table gives the instance number, the best-known lower bound computed after 24h (LB), the best-know feasible solution (best) and the associated percentage error with respect to LB (%error best/LB), the computing time needed to get the best solution (time, in CPU sec.s), and then the percentage error with respect to LB and best at the various time limits.

Comparing Table 6.7 with the performance of the MILP model alone (Table 6.6), the impact of using a matheuristic framework is clear. Using this approach, indeed, we are able to provide feasible high quality solutions for all the instances, even the more difficult ones that were unsolvable earlier. According

to Table 6.7, our method is able to solve 15 out of 29 instances to proven optimality (within the 0.01% tolerance), in many cases within few minutes. After just 1 minute, the method provides good solutions for all but 7 instances (all but 2 after 5 minutes, and all but 1 after 10 minutes). Very good (often provably optimal) solutions are available after 30 minutes for all the 29 instances in our testbed. After 1 hour the current solution value is, on average, just 0.03% from the best - known value, and 1.47% from the lower bound.

Figure 6.5: Heuristic solution value vs computing time for instance n. 3.

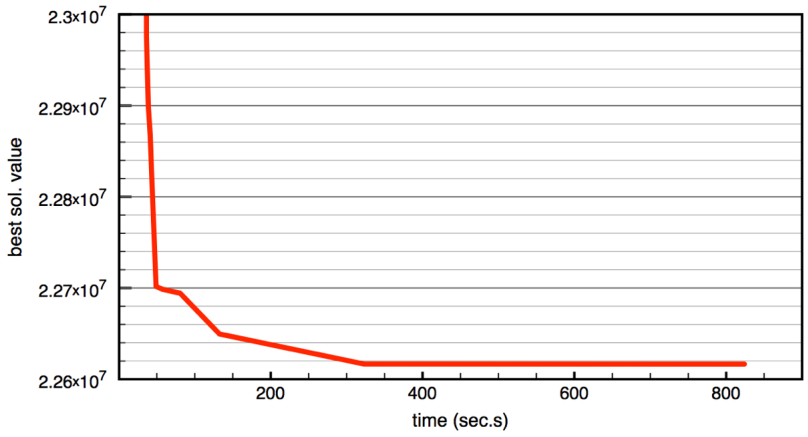
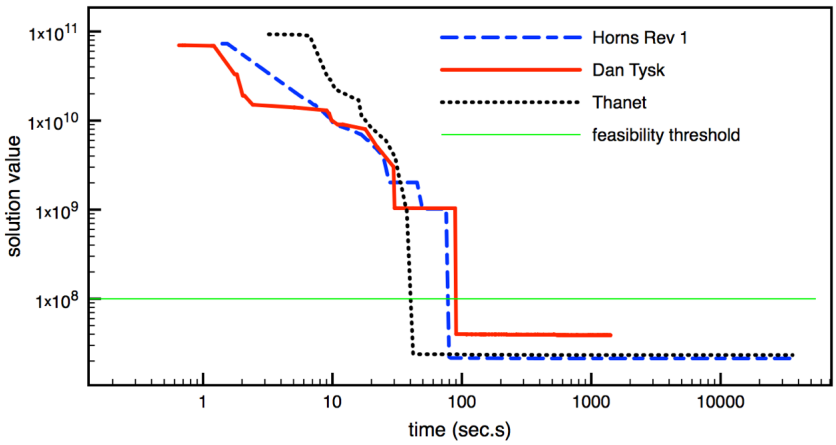


Figure 6.5 plots the best solution value over time for a typical run (instance n. 3). A feasible solution having objective 22,698,576.88 is found after 50 secs, within the initial matheuristic phase, while the best solution having objective 22,611,988.67 (at most 0.01% from optimum) is found after about 800 secs.

Note that our matheuristic framework is able to find feasible solutions very quickly also for the most difficult instances. In this matter, Figure 6.6 shows the evolution of the solution value for three difficult instances (2, 20 and 27) referring to three different real-world big parks (Horns Rev 1, Dan Tysk and Thanet). These instances were unsolvable using the MILP solver alone, but using our matheuristic approach a feasible solution is found in about 100 seconds.

Figure 6.6: Heuristic solution value vs computing time for instances n. 2, 20 and 27. The green line indicates the feasibility threshold: solutions under the green line are feasible (connected) solutions. These three instances, unsolvable with the MILP solver alone, now reach a feasible solution in less than 100 seconds.

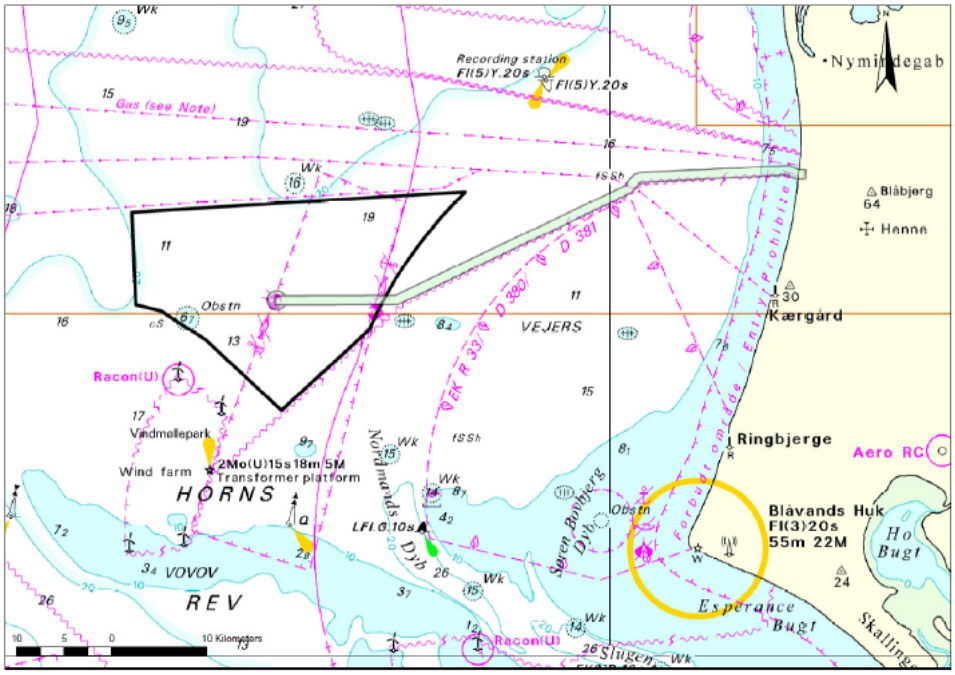


6.5 Discussion of a real case

In the previous sections we discussed the methodology we used to optimize offshore cable routing and its effectiveness for real-world cases. In what follows, we would like to give an example of the impact of considering losses on the cable layout solution. This is a major result from a commercial perspective since, having a cable route optimized considering power losses, can translate into huge savings. Here, we refer to a new wind farm, Horns Rev 3 (HR3) that is still under construction (Energinet.dk, 2013). The planned Horns Rev 3 is an area of approximately $160km^2$ in the eastern North Sea, 10-20km north of Horns Rev.

Energinet.dk has designed the substation for the future offshore wind farm as well as the connections with the coast. Therefore, the position of the substation and its characteristics are fixed. We considered a preliminary layout provided by Vattenfall with 50 8MW turbines, where turbine positions are fixed. The project team already decided that the following types of cable must be used for our tests:

Figure 6.7: Location of Horns Rev 3 (Energinet.dk, 2013)



- type 1 can have up to 3 turbines connected, a price of 180 €/m, a resistance of 0.13 Ohm/km, and insulation losses of 111 W/km.
- type 2 can have up to 4 turbines connected, a price of 360 €/m, a resistance of 0.04 Ohm/km, and insulation losses of 109 W/km.

The substation can be connected to only 12 cables, therefore we need at least 2 cables of 5 turbines each. For this reason we were allowed to overload two of the cables of type 2 (in this case it would carry 5 turbines each). Note that these overloaded cables need particular systems to monitor their temperature, which are expensive, therefore we had to limit their use (they can be used only two times).

We considered an installation price fixed at 260 €/m for both types of cable. The problem has been studied considering 13 real-world wind scenarios from HR3 with 8MW turbines. From this data, we compute the cost of cables considering losses according to (6.18). Considering cable prices, installation, cable

losses in 25 years and insulation losses, the equivalent prices for each type of cables are shown in Table 6.8.

Table 6.8: Costs for cable type 1 (considering power losses over 25 years) supporting 1, 2 or 3 turbines respectively. Costs for cable type 2 (considering losses over 25 years) supporting 4 or 5 turbines respectively

	1wtg	2wtg	3wtg	4wtg	5wtg
cable 1	470.9	561.3	711.87		
cable 2				786.5	879.7

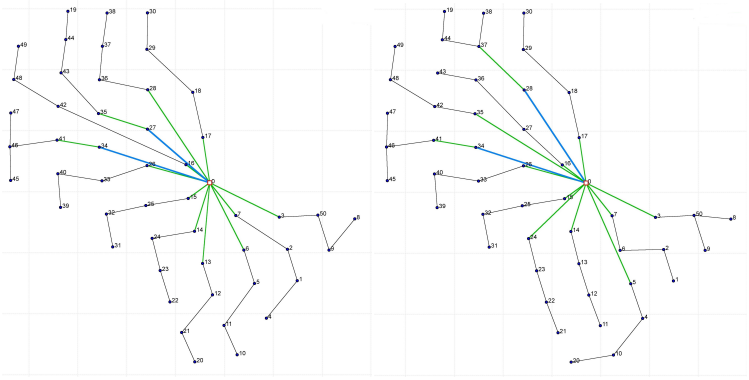
We ran our matheuristic with the original cable prices and installation costs (without losses) according to model (6.1) – (6.13). The solution found is therefore an optimized cable route from an immediate investment perspective, the so called CAPital EXpenditure (CAPEX). When we considered the new prices of Tables 6.8, we obtained a different layout: this is the optimal layout from a long term perspective, and is the one that would allow a bigger revenue in 25 years. The two layouts are compared in Figure 6.8.

In this case the long-term-optimized layout costs around 370K€ more at construction time but after 25 years this amount is paid back and another 891K€ (net present value) are saved.

One could also analyze the structure of this long-term-optimized solution compared with the CAPEX optimized one. First, in the layout optimized considering losses (second plot in Figure 6.8) there are more nodes with in-degree larger than 1, i.e., with more than one cable entering a turbine. In the layout we have 3 such nodes versus only one of the CAPEX optimized layout (first plot in Figure 6.8). To understand why, let us e.g. observe the connection between turbines 50, 8 and 9. Arc (8,9) is slightly shorter than arc (8,50), therefore it is more convenient from a CAPEX perspective. However, when considering the losses, we notice that in cable (9,50) of the first layout in Figure 6.8 the current from 2 turbines is flowing, therefore there is a larger loss compared to the connection (9,50) of the second layout where only one turbine is connected. All in all, this structure is more convenient from a long-term perspective because it maximizes the use of cable that carries less energy (from only one turbine).

We can finally compare the use of the different types of cable in the different solutions (Table 6.9): in the layout considering only CAPEX, 63% of the total cable length is for type 1 cable (the cheapest), while in the layout considering losses this value decreases to 59%. This reflects the fact that the smallest (i.e.

Figure 6.8: Optimized layouts for HR3: On the left is the layout optimized by considering only CAPEX costs, while on the right is the optimal layout considering cable losses over 25 years. Black cables are of cable type 1, green are of cable type 2, and blue are of cable type 2 carrying 5 turbines.



(a) CAPEX optimized layout *(b) Long-term optimized layout*

the cheapest) cable has the highest losses.

Table 6.9: Use of different types of cable in the two solutions. The first table refers to the CAPEX optimized solution, where 63% of the total cable length is in smaller (less expensive) cables. The second table refers to the optimized layout considering losses: the use of thicker cables (less sensible to losses) increased.

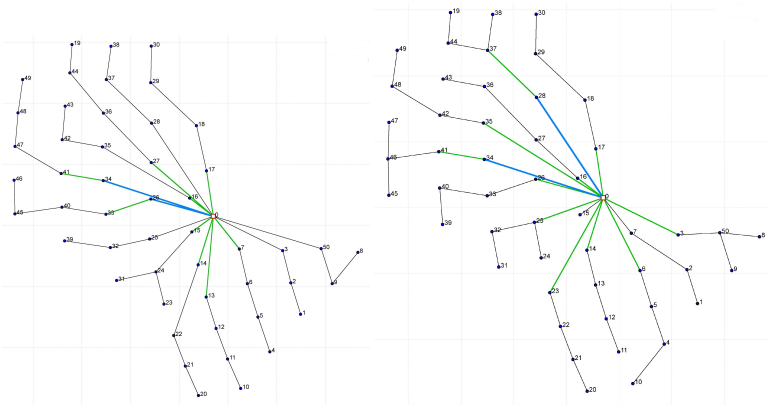
CAPEX optimized layout		
type	length [m]	% of the total
1	57149	63
2	33387	37

Long-term optimized layout		
type	length [m]	% of the total
1	52715	59
2	37130	41

This real-world test case was particularly limited by the number of connections to the substation, therefore we re-ran the cable routing optimization for the same site but imposing a maximum of 18 connections (instead of 12). Of course, to physically add connections can significantly increase the substation price, but here we just want to test the optimizer behaviour, so we suppose to have a substation already built with the extra connections. In this test the optimization is more free to play with the different cable types.

The results of the optimization considering CAPEX prices only, as well as those considering losses over 25 years (same prices as in Table 6.8) are shown in Figure 6.9.

Figure 6.9: Optimized layouts for HR3 with maximum 18 connections to the substation: plot on the left is the optimal layout considering only CAPEX costs while on the right is the layout optimized considering cable losses over 25 years. Black cables represent cable type 1, green are cable type 2 and blue are cable type 2 overloaded.



(a) CAPEX optimized layout *(b) Long-term optimized layout*

The new layouts confirm the tendencies already discussed: the layout optimized considering losses has more nodes with higher degree than the other, and is using more expensive cables. Table 6.10 shows the use of different types of cable in the two new solutions: in this case there is a difference of 15% in the use of the different kinds of cable.

Table 6.10: Use of different types of cable in the two solutions. The first table refers to the CAPEX optimized solution, where 74.5% of the total cable length is in smaller (less expensive) cables. The second table refers to the layout optimized by considering losses: the use of thicker cables (less sensible to losses) increased.

CAPEX optimized layout		
type	length [m]	% of the total
1	68910	74.5
2	23584	25.5

Long-term optimized layout		
type	length [m]	% of the total
1	53447	59
2	36693	41

In the 18 connections case the long-term layout costs around 1 324 K€ more at construction time but in 25 years this amount is paid back and another 1 156K€ (net present value) are saved. All in all, these results show the importance of considering losses when designing the inter-array cable routing.

6.6 Conclusions

In this paper we have introduced a new MILP model for optimal cable routing in offshore wind farms. A main novelty of our model is its capability of taking both installation costs and power losses into full account. We have also developed a new matheuristic framework in order to have a practical optimization tool to be used on difficult real cases. Thanks to the collaboration between Vattenfall and DTU, we have been able to describe the problem as it appears in real applications, and to validate our results on real-world instances that have been made publicly available for benchmarking. Using a sound matheuristic framework, for most of our instances we have been able to produce extremely good solutions in about 15 minutes of computing time on a standard PC.

Finally, we have fully analyzed a real wind farm whose data was provided by Vattenfall, namely Horns Rev 3, to quantify the commercial impact of considering power losses in the cable routing design. We have shown that savings in the order of millions €s can be achieved in a wind park lifetime. These kinds of highly-optimized layouts cannot be produced by a manual operator, due to the complexity of the corresponding design problem. In addition, evaluating the impact of losses on a long term perspective and understanding how the layout should be changed in order to reduce them, represents an extremely valuable analysis for a company, that could not have been carried out otherwise.

References

4cOffshore (2015). *website*. `http://www.4coffshore.com/offshorewind/`. Accessed: 2015-01-30.

Archer, R., G. Nates, S. Donovan, and H. Waterer (2011). "Wind Turbine Interference in a Wind Farm Layout Optimization – Mixed Integer Linear Programming Model". In: *Wind Engineering* 35, no.2, pp. 165–178.

Bauer, J. and J. Lysgaard (2015). "The offshore wind farm array cable layout problem: a planar open vehicle routing problem". In: *Journal of the Operational Research Society* 66.3, pp. 360–368.

Berzan, C., K. Veeramachaneni, J. McDermott, and U. O. Reilly (2011). "Algorithms for cable network design on large-scale wind farms". In: *Tech. Rep. Tufts University*.

Cerveira, A., A. D. Sousa, E. J. S. Pires, and J. Baptista (2016). "Optimal Cable Design of Wind Farms: The Infrastructure and Losses Cost Minimization Case". In: *IEEE Transactions on Power Systems* 31.6, pp. 4319–4329.

Cordone, R. and F. Maffioli (2004). "On the complexity of graph tree partition problems". In: *Discrete Applied Mathematics 134*, pp. 51–65.

Dutta, S. (2012). "Data Mining and Graph Theory Focused Solutions to Smart Grid Challenges". MA thesis. University of Illinois.

Dutta, S. and T. J. Overbye (2011a). "A Clustering Based Wind Farm Collector System Cable Layout Design". In: *Power and Energy Conference at Illinois (PECI)*, pp. 1–6.

Dutta, S. and T. J. Overbye (2011b). "A Clustering based Wind Farm Collector System Cable Layout Design". In: *Power and Energy Conference at Illinois (PECI)*, pp. 1–6.

Energinet.dk (2013). "Technical Project Description for the large-scale offshore wind farm (400MW) at Horns Rev 3, Document no. 13-93461-267". In:

Fagerfjall, P. (2010). "Optimizing Wind Farm Layout - More Bang for the Buck Using Mixed Integer Linear Programming". MA thesis. Goteborg, Sweden: Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University.

Fischetti, M. and A. Lodi (2011). "Heuristics in Mixed Integer Programming". In: *Wiley Encyclopedia of Operations Research and Management Science (James J. Cochran ed.) Vol. 8*. John Wiley and Sons, pp. 738–747.

Fischetti, M. and M. Monaci (2015). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics*. ISSN: 1381-1231.

Fischetti, M. and D. Pisinger (2017). "Optimizing wind farm cable routing considering power losses". In: *European Journal of Operational Research*. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.07.061. URL: http://www.sciencedirect.com/science/article/pii/S037722171730704X.

Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

González, J. S., M. B. Payán, J. M. R. Santos, and F. González-Longatt (2014). "A review and recent developments in the optimal wind-turbine micro-siting problem". In: *Renewable and Sustainable Energy Reviews* 30, pp. 133–144.

González-Longatt, F. M. and P. Wall (2012). "Optimal Electric Network Design for a Large Offshore Wind Farm Based on a Modified Genetic Algorithm Approach". In: *IEEE Systems Journal* 6.1, pp. 164–172.

Hansen, P., V. Maniezzo, and S. Voß (2009). "Special issue on mathematical contributions to metaheuristics editorial". In: *Journal of Heuristics* 15.3, pp. 197–199.

Hertz, A., O. Marcotte, A. Mdimagh, M. Carreau, and F. Welt (2012). "Optimizing the Design of a Wind Farm Collection Network". In: *INFOR* 50.2, pp. 95–104.

Kis-orca (2015). *website*. http://www.kis-orca.eu/downloads/. Accessed: 2015-01-30.

Kusiak, A. and Z. Song (2010). "Design of wind farm layout for maximum wind energy capture". In: *Renewable Energy* 35.3, pp. 685–694.

Li, D., C. He, and Y. Fu (2008). "Optimization of Internal Electric Connection System of Large Offshore Wind Farm with Hybrid Genetic and Immune Algorithm". In: *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT2008)*, pp. 2476–2481.

Qi, W., Y. Liang, and Z.-J. M. Shen (2015). "Joint Planning of Energy Storage and Transmission for Wind Energy Generation". In: *Operations Research* 63.6, pp. 1280–1293.

Samorani, M. (2013). "The Wind Farm Layout Optimization Problem". In: *Handbook of Wind Power Systems*. Energy Systems, pp. 21–38.

Zhao, M., Z. Chen, and F. Blaabjerg (2009). "Optimisation of Electrical System for Offshore Wind Farms via Genetic Algorithm". In: *IET Renewable Power Generation 3* 3.2, pp. 205–216.

Chapter 7

On the impact of power losses in the design of offshore wind farm cable routing

Martina Fischetti[a] · David Pisinger[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]Management Science, Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

Publication Status: Submitted as invited paper to the Springer Book of ICORES 2017.

A reduced version of this paper, was awarded the **Best Student Paper Award** at the 6th International Conference on Operations Research and Enterprise Systems (ICORES), Porto, 2017

Reading Instructions: The focus here is on the impact of using the model of Fischetti and Pisinger, 2017b in reality. If you read Chapter 6, you can go directly to Section 7.4

Abstract: Wind energy is a field of main importance in the transition away from fossil fuels. In order to achieve this goal, reducing production cost of wind energy is of primary importance, especially for offshore wind parks. In the present paper we illustrate optimization models to achieve this goal for the cable routing problem. In particular we focus on the economical impact of considering power losses in the optimization. The resulting optimization problem considers both minimizing immediate costs (CAPEX) and minimizing costs due to power losses in the park lifetime. Thanks to the close collaboration with a leading energy company, we have been able to conduct different what-if analyses on a set of existing wind parks. Having a fast and reliable tool to optimize cable routing considering or not power losses, we have been able, for the first time, to quantify the impact of these kinds of decisions at design phase. Our results illustrates the importance of considering power losses already at the design phase, as well as the importance of having a sophisticated optimization tool, compared with the traditional manual design.

Keywords: Mixed Integer Linear Programming, Offshore Wind Parks, Green Energy, Cable Routing, Cable Losses

7.1 Introduction

Wind power is a leading technology in the transition to green sources of energy. Having a yearly market growth of 15-20%, it is however necessary to face new challenges on a market that is more and more competitive. According to González et al., 2014 the expenses for electrical infrastructure of a offshore wind farm account for 15-30% of the overall initial costs. Therefore, high-level optimization in this area is a key factor. Cable layout is a problem of great interest in many companies and it is typically solved only manually. Different types of cable layout problems can be addressed: in this paper we study the

inter array cable optimization, i.e., the optimal routing to connect offshore turbines and to collect their energy in one or more substations. In particular, we focus on the impact of considering power losses already in the design phase. While the energy flows through a cable, indeed, part of it gets lost due to the intrinsic resistance of the cable. An optimized selection of the cable structure and the cable type, can reduce the amount of these losses.

The main scope of the inter-array cable routing is to collect the power production of turbines in offshore substations. To do that, each turbine must be connected to one substation through a loop-free path. The inter-array cable routing optimization problem consists in finding the cable connection that minimizes the associated cost. Since different cables with different costs, capacities and resistances are available on the market, the task is not only to find the turbines to be directly connected, but also to choose appropriate cable types to minimize losses.

Wind park cable routing optimization has obtained considerable attention in the last years. Due to the large number of constraints and the intrinsic complexity of the problem, many studies (i.e. Dutta and Overbye, 2011; González-Longatt and Wall, 2012; Li et al., 2008; Zhao et al., 2009) preferred to use ad-hoc heuristics. Only a few papers used Mixed Integer Linear Programming (MILP), notably Bauer and Lysgaard, 2015; Fagerfjall, 2010; Dutta, 2012; Berzan et al., 2011; Hertz et al., 2012; Cerveira et al., 2016; Pillai et al., 2015. A MILP approach boosted with heuristics (a so-called matheuristic approach) to deal with large-scale wind parks in an acceptable time has been recently proposed in Fischetti and Pisinger, 2017b. The present work is based on Fischetti and Pisinger, 2017b but focuses more on real applications of the optimization model and on its economical impact. Several variants of the problem have been proposed in the literature. To the best of our knowledge only Cerveira et al., 2016 has considered power loss in cables. However, Cerveira et al., 2016 does not take into account variable cable loads due to fluctuating wind. Bauer and Lysgaard, 2015 proposes an Open Vehicle Routing approach for this problem adding the planarity constraints on the fly. In this Open Vehicle Routing version of the problem, only one cable can enter a turbine, even if this is often not the case in the reality. In Bauer and Lysgaard, 2015, the possibility of branching cables in the turbines (as we are doing), is mentioned as a future work. However, the substation limits, that could be a major constraint in practical applications, are not considered in Bauer and Lysgaard, 2015. Different approaches for the cable network design are provided in Berzan et al., 2011. The suggested approach is a divide-and-conquer heuristic based on the idea of dividing the big circuit problem into smaller circuit ones. They also propose a

MILP model, but it cannot deal with more than 11 turbines. In Hertz et al., 2012 the cable layout problem for onshore cases is studied.

Thanks to the collaboration with a leading energy company it has been possible to build a detailed model including a majority of the constraints arising in practical applications, and to evaluate the savings of optimized layouts on real cases. First, the energy flow is unsplittable (so the flow leaving a turbine must be supported by a single cable) and the flow in each cable cannot exceed its capacity. Secondly, also the substations that collect the energy have some limitations. In particular, each substation has a maximum number of electric connections, i.e., a maximum number of cables that can be connected to it. Moreover, cable crossings should be avoided. Cable crossing is not impossible in principle, but is highly not recommended in practice. Building one cable on top of another is, indeed, more expensive and increases the risk of cable damages. Therefore it is important for a model to take this planarity constraint into account. We used a Mixed Integer Linear Programming (MILP) approach with ad-hoc heuristics to solve difficult instances of this problem Fischetti and Pisinger, 2017b. The resulting optimization tool has been validated by company experts, and is now routinely used by the planners.

The main contribution of the present paper is to analyse how the inter-array cable routing of real-world wind farms can be improved by using modern optimization techniques. A particularly challenging aspect in the cable routing design is to understand if one could limit power losses by optimizing cable routing. As a general rule, cables with less resistance are also more expensive, therefore we would like to make a proper trade-off between investments and cable losses. We formulate the optimization problem with immediate costs (CAPEX) and losses-related costs as two separate goals. The two objectives can be merged into a single objective by proper weighing of the two parts. The weighing factor can be considered fixed or can vary: this makes it possible to perform various what-if analyses to evaluate the impact of different preferences (i.e. weighing factors). This analysis is important in cases where a positive pay-back is demanded within a short time horizon, or where liquidity problems hinder choosing the best long-term solution. We report a study of both approaches on a set of real-world instances.

In our computation of power losses, we show that wind scenarios can be handled efficiently as part of data preprocessing, resulting in a MILP model of tractable size. Tests on a library of real-life instances proved that substantial savings can be achieved.

Our paper is organized as follows: Section 7.2 describes our MILP model, first

presenting a basic model and then improving and extending the formulation. In particular, we show how to model power losses, and propose a precomputing strategy that is able to handle this non-linearity efficiently, thus avoiding sophisticated quadratic models that would make our approach impractical. In Section 7.3 we describe how to handle large size instances using a matheuristic approach. Section 7.4 compares our optimized solutions with an existing cable layout for a real wind farm (Horns Rev 1), showing that millions of euro can be saved. Section 7.5 is dedicated to various what-if analyses. In particular, Subsection 7.5.1 describes the real-world wind farms that we considered in our tests, while Subsection 7.5.2 shows the results of our optimization on a testbed of real-world cases, reporting the impact of considering power losses for all the instances. Section 7.6 is dedicated to different analyses on the weighting factor: Subsection 7.6.1 analyses the impact of different return-of-investment requirements on the cable routing costs, while Subsection 7.6.2 studies the impact of considering price fluctuations on the market. Some conclusions are finally addressed in Section 7.7.

The present paper is an extended version of the conference paper Fischetti and Pisinger, 2017a from the same authors.

7.2 Mathematical models

In order for this paper to be self-contained, we start by reviewing the MILP models and algorithms we proposed in Fischetti and Pisinger, 2017b; the interested reader is referred to the given paper for details.

7.2.1 Basic model

We assume that the location of the turbines has already been defined. We wish to find an optimal cable connection between all turbines and the given substation(s), minimizing the total cable costs. The optimization problem considers that:

- the energy flow leaving a turbine must be supported by a single cable;
- the maximum energy flow (when all the turbines produce their maximum) in each connection cannot exceed the capacity of the installed cable;

- different cables, with different capacities, costs and impedances, can be installed;
- cable crossing should be avoided;
- a given maximum number of cables can be connected to each substation;
- cable losses (dependent on the cable type, the cable length and the current flow through the cable) must be considered.

We will first model the problem without cable losses and then discuss in Subsection 7.2.2 how to efficiently express these latter constraints. We model turbine positions as nodes of a complete and loop-free directed graph $G = (V, A)$ and all possible connections between them as directed arcs. Some nodes correspond to the substations that are considered as the roots of the trees, being the only nodes that collect energy. Let P_h be the power production at node h. We distinguish between two different types of node: V_T is the set of turbine nodes, and V_0 is the set of substation nodes. Let T denote the set of different cable types that can be used. Each cable type t has a given capacity k_t and unit cost u_t, representing the cost per meter of the cable (CAPEX). Arc costs can therefore be defined as $c^t_{i,j} = u_t dist(i,j)$ for each arc $(i,j) \in A$ and for each type $t \in T$, where $dist(i,j)$ is the Euclidean distance between turbine i and turbine j. In our model we use the continuous variables $f_{i,j} \geq 0$ for the flow on arc (i,j). The binary variables $x^t_{i,j}$ define cable connections as

$$x^t_{i,j} = \begin{cases} 1 & \text{if arc } (i,j) \text{ with cable type } t \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, variables $y_{i,j}$ indicate whether turbines i and j are connected (with any type of cable). Note that variables $y_{i,j}$ are related to variables $x^t_{i,j}$ as $\sum_{t \in T} x^t_{i,j} = y_{i,j}$. The overall model can be stated as follows Fischetti and Pisinger, 2017b:

$$\begin{aligned}
\min \quad & \textstyle\sum_{i,j \in V} \sum_{t \in T} c^t_{i,j} x^t_{i,j} && && (7.1)\\
\text{s.t.} \quad & \textstyle\sum_{t \in T} x^t_{i,j} = y_{i,j}, && i,j \in V : j \neq i && (7.2)\\
& \textstyle\sum_{i : i \neq h} (f_{h,i} - f_{i,h}) = P_h, && h \in V_T && (7.3)\\
& \textstyle\sum_{t \in T} k_t x^t_{i,j} \geq f_{i,j}, && i,j \in V : j \neq i && (7.4)\\
& \textstyle\sum_{j : j \neq h} y_{h,j} = 1, && h \in V_T && (7.5)
\end{aligned}$$

$$\sum_{j:j\neq h} y_{h,j} = 0, \qquad h \in V_0 \tag{7.6}$$
$$\sum_{i\neq h} y_{i,h} \leq C, \qquad h \in V_0 \tag{7.7}$$
$$x^t_{i,j} \in \{0,1\}, \qquad i,j \in V, t \in T \tag{7.8}$$
$$y_{i,j} \in \{0,1\}, \qquad i,j \in V \tag{7.9}$$
$$f_{i,j} \geq 0, \qquad i,j \in V, j \neq i. \tag{7.10}$$

The objective function (7.1) minimizes the total cable layout cost. Constraints (7.2) impose that only one type of cable can be selected for each built arc, and defines the $y_{i,j}$ variables. Constraints (7.3) are flow conservation constraints: the energy (flow) exiting each node h is equal to the flow entering h plus the power production of that node (except if the node is a substation). Constraints (7.4) ensure that the flow does not exceed the capacity of the installed cable, while constraints (7.5) and (7.6) impose that only one cable can exit a turbine and none can exit the substations (tree structure with root in the substations). Finally, constraints (7.7) impose the maximum number of cables (C) that can enter each substation.

In order to model no-cross constraints we need a constraint for each pair of crossings arcs, i.e. a very large number of constraints. We have, therefore, decided to generate them on the fly, as also suggested in Bauer and Lysgaard, 2015. In other words, the optimizer considers model (7.1) - (7.10) and adds the following new constraints whenever two established connections (i,j) and (h,k) cross

$$y_{i,j} + y_{j,i} + y_{h,k} + y_{k,h} \leq 1. \tag{7.11}$$

The reader is referred to Fischetti and Pisinger, 2017b for stronger versions of those constraints. Using this approach, the number of non-crossing constraints actually added to the model decreases dramatically, making the model faster to solve. As presented, the model is able to deal with small size instances only. In order to produce high quality solutions in an acceptable amount of time also for large-scale instances, a "matheuristic" framework (as the one proposed in Fischetti and Pisinger, 2017b) can be used on top of this basic model. We refer to Section 7.3, for more details.

7.2.2 Cable losses

In this section we review an extension of the previous model taking cable losses into account (still from Fischetti and Pisinger, 2017b). Consider a generic cable of type t under wind scenario s. Power losses increase with the square of the

current $g_{i,j}^{t,s}$, according to the formula:

$$3R^t \cdot dist(i,j)(g_{i,j}^t)^2 \tag{7.12}$$

where R^t is the electrical resistance of the 3-phase cable of type t, in Ω/m. Decision variable $g_{i,j}^{t,s}$ obviously depends on the considered wind scenario. As a consequence, dealing with equation (7.12) directly in the model, would imply dealing with non-linearities over multiple scenarios. Nevertheless, (7.12) can be simplified if we assume that all the turbines in the park have the same power production under the same wind scenario. This is a fair assumption since typical parks are constructed by using only one turbine model and wake effect is not usually considered in electrical studies. Under this assumption, the current I_s passing through a generic cable supporting f turbines (say), can be expressed as $g_{i,j}^{t,s} = fI^S$ where I^s is the current produced by a single turbine under scenario s. Accordingly, power loss can be expressed as a function of f, as

$$PLoss^{t,f,s} = 3R^t dist(i,j)(fI^s)^2. \tag{7.13}$$

The value $f \in 1, ...F$ is limited by the capacity of the cables. By introducing the dependency on f in our main binary variables (now $x_{i,j}^{t,f}$) we can re-write our two cost contribution as:

$$min \sum_{i,j\in V} \sum_{t\in T} \sum_{f\in F} \sum_{s\in S} \pi_s PLoss^{t,f,s} x_{i,j}^{t,f} \tag{7.14}$$

and

$$\min \quad \sum_{i,j\in V} \sum_{t\in T} \sum_{f\in F} c_{i,j}^t x_{i,j}^{t,f}, \tag{7.15}$$

where π_s is the probability of scenario s. As we have discussed earlier, minimizing losses can imply an increase of the CAPEX cost, therefore the two objective must be properly balanced. In some cases (e.g., when there is no limit on the CAPEX) they can be merged, by using a converting factor for the loss-related term: this is the estimated cost for each MW of production lost over the wind farm lifetime (Net Present Value). This value (denoted K) is an input value, that the designer can set to the desired project-specific value. The merged objective function, now expressed in €, is then:

$$\min \sum_{i,j\in V} \sum_{t\in T} \sum_{f\in F} c_{i,j}^t x_{i,j}^{t,f} + K \sum_{i,j\in V} \sum_{t\in T} \sum_{f\in F} \sum_{s\in S} \pi_s PLoss^{t,f,s} x_{i,j}^{t,f}. \tag{7.16}$$

The new set of variables $x_{i,j}^{t,f}$ can actually be handled implicitly in a pre-processing phase, without changing the original model (7.1)-(7.10), according

to the following idea. We consider the basic model (7.1)–(7.10) without cable losses on a modified instance where each cable type is replaced by a series of "subcables" with discretized capacity and modified cable cost taking both CAPEX and revenue losses due to cable losses into account.

Nearly all wind farms are designed for only one turbine type, hence the maximum power production P_h of each turbine can be normalized to 1, meaning that we can express the cable capacity as the maximum number of turbines supported. Consider a certain cable type t that can support up to k_t turbines. We replace it by k_t "subcable" types of capacity $f = 1, 2, ..., k_t$ whose unit cost is computed by adding both cable/installation unit costs (u_t) and loss costs (denoted as $loss^{t,f}$) considering the current produced by exactly f turbines. Note that such unit costs increase with f, so the optimal solution will always select the subcable type f supporting exactly the number of turbines connected, hence the approach is correct.

The above approach allows us to easily consider multiple wind scenarios without affecting the model size. This is obtained by precomputing the subcable unit costs by just considering a weighted average of the loss unit cost under different wind scenarios (and hence different current productions). To be more specific, we can now precompute the value

$$loss^{t,f} = 3R^t K \sum_{s \in S} \pi_s (fI^s)^2 \quad , \tag{7.17}$$

where π_s is the probability of scenario s and I^s is the current produced by a single turbine under wind scenario s. We refer to the next subsection for a more detailed example of how cable costs are pre-processed when considering losses. As said, K is a factor to estimate the value (in €) of a MW loss, and can be computed as $K = K_{euro} \cdot 8760$ where K_{euro} is the NPV for a MW/h production over the park lifetime, and 8760 is the number of hours in a year. Notice that K_{euro} acts as a weighing factor between the two objectives: minimize CAPEX costs versus minimize losses. In practice, this value is site-specific so it is given by the business team of the specific farm. It takes into account the expected cost of energy and the lifetime of the park. In Section 7.6.1 we will sketch a sensitivity analysis on the variation of this parameter, looking in particular at the effect of considering a shorter return of investment for the park. In general, we will consider a unique K_{euro} that does not follow the variations of the spot price: wind parks, indeed, commonly operate at a protected and fixed price for most of their lifetime (at least in Denmark). In Section 7.6.2 we will consider the case of using market prices, i.e. having a different K_{euro} for different wind scenarios. We will analyse the impact of considering price fluctuation on the losses optimized solution on real-world instances.

7.2.3 Loss pre-computation

In this section we illustrate the pre-computing strategy proposed in the previous session, using a concrete example from the real wind park Horns Rev 1. The park consists of 80 2MW turbines and is located about 15 km from the Danish shore. This park will be used as one of our test cases in Sections 7.4 and 7.5.

Cable sets can differ in cable cross section or in voltage (33kV or 66kV generally), which reflects in different capacities and resistances. The set of most adequate cables is selected by the electrical specialists in the company. Of course, different cable types can lead to different optimal layouts, as we will see in Section 7.5.

Let us suppose that we are given a set of two cables: the cheapest one can support 10 2MW turbines and the most expensive 14 turbines. This set of cables will be indicated as cb05 in Section 7.5. We are provided with the following table, that reports the characteristics of the two cable types (including installation costs).

Table 7.1: Cable information for cb05 Fischetti and Pisinger, 2017a.

cables	type	n. of 2MW turb.	resistance [Ohm/km]	cost [€/m]	install. cost [€/m]
cb05	1	10	0.13	180	260
	2	14	0.04	360	260

If we want to optimize on CAPEX costs only, we just need to input to the model the capacity of each cable type and its overall cost (cable price plus installation cost). In this case, for example, this would be:

- type 1: supports up to 10 turbines with a unit cost of 440 €/m
- type 2: supports up to 14 turbines with a unit cost of 620 €/m.

Third column of Table 7.2 shows how the model will compute the unit price (CAPEX only) depending on the number of turbines connected.

Let us now consider losses using the strategy of Subsection 7.2.2. As we discussed earlier, the power loss in a cable depends on the current passing through it. Since only a discrete number of turbines can be connected to each cable

path, we can express the current as a function of the number f of turbines connected without any loss of precision in the result.

Still referring to equation (7.17), the losses depend also on the wind statistics in the site. We can define a wind scenario (s) as a wind speed and its probability to occur (π_s). At a given wind speed, a given turbine will produce a specific current (I_s).

Wind scenarios can be defined in different ways. In this paper we used both real measurements and scenarios derived from Weibull distributions for the specific sites. For the Horns Rev 1 case we are considering, we had real measurements from the site, i.e., a wind speed sample each 10 minutes for 10 years. We grouped all these samples in wind-speed bins of 1m/s, obtaining 25 wind scenarios (from 1 to 25 m/s). The probability of each scenario was obtained looking at the frequency of the specific wind speed over all the samples. In our tests we decided to bin our data every 1 m/s, following the practice in electrical losses computations. However this should not be considered a limit: since the wind scenarios are handled in the pre-processing phase, the number of scenarios does not affect the size of the final optimization model.

Having computed I_s and π_s according to the scenario definition, power losses can now be calculated. Parameter $K_{euro} = 690$ €/MWh was computed by the company experts for a wind park lifetime of 25 years, while resistance R_t is defined according to Table 7.1. Using equation (7.17), the cost for power losses $loss^{t,f}$ can be now precomputed. As shown in (7.16), the cost considered in the objective for each cable connection will need to include the CAPEX costs (u_t) and the contribution from losses ($loss^{t,f}$). Therefore the final input to the optimization tool for Horns Rev 1 with cb05, will be as shown in the fourth column of Table 7.2.

A comparison between the last two columns of Table 7.2 shows the impact of considering losses on cable prices. While from a installation perspective the cost for each cable type is fixed, it now varies depending on how many turbines are connected. As we will see, this can have a significant impact on the optimal cable routing.

7.3 Matheuristics

In a practical setting, one would like to find high-quality solutions in short computing time, making it possible to experiment with different settings. This

Table 7.2: Precomputed cable prices for cable cb05 (including installation costs) for Horns Rev 1. First column indicates the cable type, second column indicates the number of turbines supported f. Third column indicates the CAPEX costs, while fourth column reports prices with also power losses costs included.

cable type	n. of 2MW turb. supported	CAPEX cost [€/m]	cost with losses [€/m]
1	1	440	441.16
	2	440	442.71
	3	440	445.27
	4	440	448.87
	5	440	453.50
	6	440	459.15
	7	440	465.83
	8	440	473.54
	9	440	482.28
	10	440	492.04
2	11	620	639.77
	12	620	643.41
	13	620	647.36
	14	620	651.63

could be useful, for example, for what-if analyses considering different cables from different manufactures, or to evaluate the effect of different design choices (as we will do in Section 7.5). In some difficult cases, model (7.1)-(7.10) could require long computing time before producing even the first feasible solution. On the other hand, due to the intrinsic structure of MILP solvers, having a first solution as soon as possible in the branch-and-bound tree could significantly speed up the overall resolution. In Fischetti and Pisinger, 2017b we used MILP-based heuristics on a relaxed version of the model to quickly produce first solutions for the MILP solver. The relaxed model and the matheruistics applied on it, are next outlined for the sake of completeness.

7.3.1 A relaxed model

Model (7.1) - (7.10) can be relaxed to find feasible solutions faster. This can be obtained by allowing for disconnected solutions, that are penalized by high costs.

To be more specific, we introduce a new variable, $l_h \geq 0$, that indicates the loss at the node h.The cost of a unit loss is fixed to M, a large positive constant

greater than all the prices involved in the optimization (we used 10^9). This is to ensure that a connected solution would always have a lower cost compared with a disconnected one.

The relaxed model is then obtained from (7.1) - (7.10) by replacing (7.1) with

$$\min \quad \sum_{i,j\in V} \sum_{t\in T} c^t_{i,j} x^t_{i,j} + \quad \sum_i M l_i \tag{7.18}$$

and (7.3) with

$$\sum_{i:i\neq h} (f_{h,i} - f_{i,h}) = P_h - l_h \quad h \in V_S \cup V_T. \tag{7.19}$$

A MILP solver applied to the relaxed model is typically able to find, in a few seconds, a feasible (possibly disconnected) first solution and to quickly proceed in the tree enumeration to discover better and better ones. Therefore, the relaxed model is used in our experiments.

7.3.2 Matheuristics based on the relaxed model

As its name suggests, a matheuristic is the hybridization of mathematical programming with metaheuristics. The idea presented in what follows is to use the relaxed model powered up by the use of a metaheuristic. We refer the interested reader to Boschetti et al., 2009; Hansen et al., 2009; Fischetti and Fischetti, 2016, for a more general treatment of the subject. The basic idea of our matheuristic is to restrict the number of variables in the optimization by temporary fixing some arcs of the best solution found so far, and re-optimize on the remaining arcs. In other words, given a feasible solution y^* of the relaxed model, we fix to 1 some of the y variables with $y^*_{i,j}$. Note that in our problem, fixing some arcs implies to exclude all the crossings arcs, with a drastic redaction in the dimension of the model. In order to decide which arcs to fix in the solution, we used different heuristic strategies.

Our first matheuristic works as follows: The relaxed model is solved with a short time limit and the best found solution y^* is returned. Afterwards, the arcs selected in this solution (i.e., all those variables having $y^*_{i,j} = 1$) are temporally fixed with a certain probability (e.g. 0.5). The resulting restricted problem is reoptimized on the remaining arcs, and the approach is repeated.

As already observed, every time some arcs are heuristically fixed on input, in a preprocessing phase we can forbid all possible crossing arcs (i.e. we can set to

zero all the variables related to them). This is very important for the success of our heuristic, as the restricted problems become much easier due to the fixing.

Our second matheuristic uses a similar approach, but with a more problem-related strategy to choose the fixing probability. Instead of having a fixed probability to select arcs, the probability is now related with the distance to the substation(s): the arcs closer to the substation(s) are fixed with a higher probability. The distance of an arc (i, j) to the substation(s) is defined as

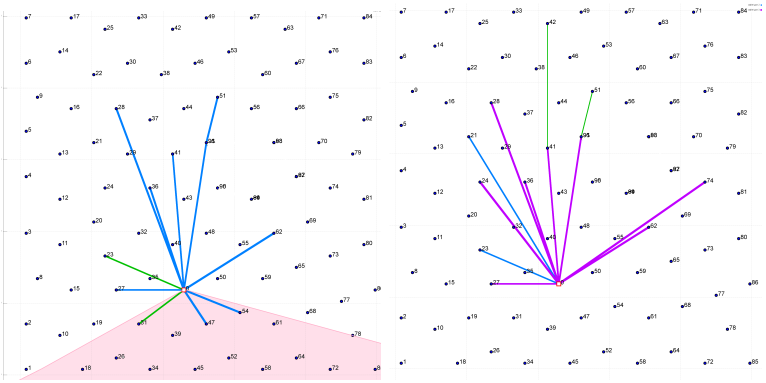
$$DIST_{i,j} = max\{min_{h \in V_0} dist(i, h), min_{h \in V_0} dist(j, h)\}.$$

Distances are normalized with respect to the longest distance in the specific test instance ($DIST_{MAX}$) and the fixing probability is computed as $1 - DIST_{i,j} / DIST_{MAX}$. In this way the arcs closer to the substation have an higher probability to be fixed and the optimization tends to focus on the more far away arcs.

The third matheuristic is specific for wind farms with only one substation. Analysing the solutions, indeed, the layout appears divided in sectors: the final layout looks as a collection of "irregular rays" connected to the substation. Our third matheuristic is therefore randomly decomposing the problem in sectors, fixing the arcs outside the sector and re-optimizing inside. To be more specific, all the nodes are ordered according to their angle with the substation. A turbine (that we will call "the seed") is randomly selected and the sector is defined by picking the next μ turbines in the ordered sequence (e.g. $\mu = 30\%$ of the total number of turbines). The arcs connecting turbines outside this sector are fixed while any arc (i, j) where i or j is in the sector, is reoptimized. The already discussed pre-processing is applied on the fix arcs and the optimization is re-run. If the new solution is improved, we select a new seed close to the previous one (i.e. in the ordered vector we pick a node in the interval [current seed -2, current seed +2], according to a normal distribution), while, if the solution is not improved, the new seed is the 5^{th} turbine after the current seed. Figure 7.1 illustrates this last heuristic. Turbines are represented as black dots, while the substation is the red square. Different cable types are represented by arcs of different colours. Some turbines are connected to other turbines or to the substation with different types of cables (in blue and green). The first plot in Figure 7.1 shows the first (disconnected) solution obtained using a MILP solver on the relaxed model with a few-second timelimit. A sector is defined on this solution (in pink in the picture). The variables y referring to arcs outside this sector are fixed to 1. This means that these connections are fixed in the next iteration (cable types are instead not fixed, meaning

that the colour of the connection in the plot can vary). The restricted MILP model (with fixed variables) is passed again to the MILP solver with a short timelimit. The second plot in Figure 7.1 shows the new solution we got from the solver. The arcs outside the sector are kept in the new solution, even if the type of cable changed, while the arcs inside the sector are not reselected in the new solution. By repeating this framework many times, the solutions quickly improve exploring different neighbourhoods.

Figure 7.1: Two consectutive iterations of the Sector matheuristic: after a short timelimit we receive a first disconnected solution from the solver (left plot). We define a sector (in pink) and we fix all the connections outside the sector (setting the corresponding y variables at 1). We pass the new (restricted) problem to the solver, that returns the solution on the second plot (right plot). We iterate the process obtaining still better solutions to warm start the MILP solver.

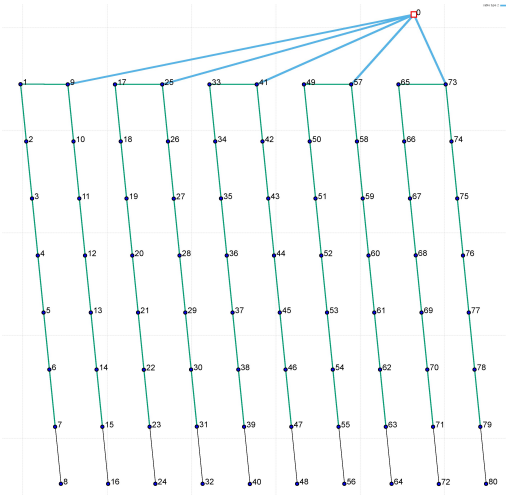


All three matheuristics are used in our tests, repeating them 5 times before starting the final MILP-solver run (without any fixing). We refer to Fischetti and Pisinger, 2017b for a computational analysis on the impact of using matheuristics techniques on the wind farm cable routing problem.

7.4 Comparison with an existing layout

We report in this section a comparison between our optimized solutions (considering and not considering losses) and the existing cable routing for Horns Rev 1, a real-world offshore park located in Denmark. Figure 7.2 shows the actual design for Horns Rev 1 (from Kristoffersen and Christiansen, 2003).

Figure 7.2: Existing cable routing for Horns Rev 1 Fischetti and Pisinger, 2017a.



Three different types of cables are used: the thinnest cable supports one turbine only, the medium supports 8 turbines, and the thickest 16. We estimated the costs and resistances of these cables based on the cable cross section. The estimated prices are 85 €/m, 125 €/m and 240 €/m, respectively, plus an estimated 260 €/m for installation costs (independent of the cable type). We ran our CAPEX optimization with the above prices obtaining the layout in Figure 7.3. The optimized layout is significantly different from the existing one. Looking at immediate costs, the optimized layout is more than 1.5 M€ less expensive. As already said, this layout is optimized only on immediate costs, nevertheless if we estimate its value in 25 years (considering losses) it is still more profitable than the existing one (by about 1.6 M€).

By optimizing cable losses, one can further increase the value in the long term. Figure 7.4 shows the optimized solution considering losses (thus optimizing the value of the cable route in its lifetime). Compared with the existing layout (Figure 7.2), this new layout is about 1.7 M€ (NPV) more profitable in 25 years, and still around 1.5 M€ cheaper at construction time.

Table 7.3 summarizes the savings of the two optimized layouts compared with the existing one, both from an immediate cost perspective and from a long-term perspective; all values are expressed in K€.

The test shows that more than one million Euros can be saved using our optimization methods on real parks. In the next section we want to focus on the

Figure 7.3: Optimized layout for Horns Rev 1 (CAPEX costs only): this layout is more than 1.5 M€ more profitable than the existing one Fischetti and Pisinger, 2017a.

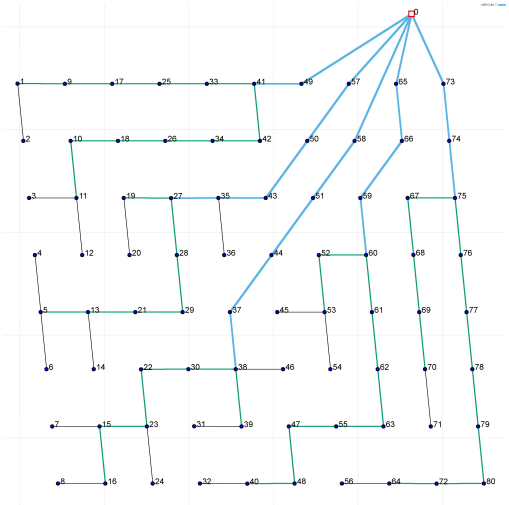


Figure 7.4: Optimized layout for Horns Rev 1 (considering losses): in the wind park lifetime this layout is estimated to be more than 1.7 M€ more profitable than the existing one Fischetti and Pisinger, 2017a.

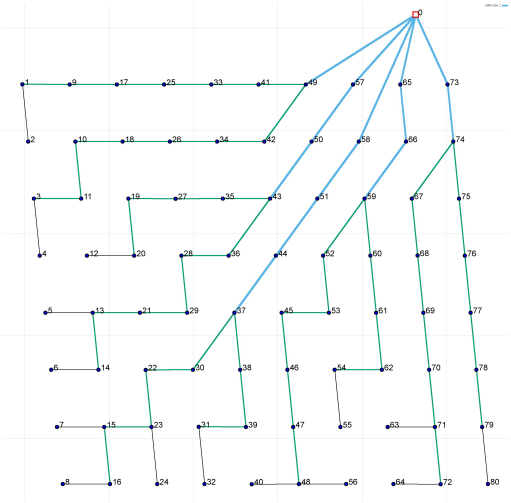


Table 7.3: Savings of optimized solutions compared with the existing cable routing for Horns Rev 1 Fischetti and Pisinger, 2017a.

	Savings [K€]	
opt mode	immediate	in 25years
CAPEX	1544	1605
lifetime	1511	1687

other big advantage of using automatic optimization tools: the possibility of performing a number of what-if analyses. To the best of our knowledge, this is the first detailed study on the impact of different design choices on the cable routing itself and on its impact on immediate costs (CAPEX) and long term costs.

7.5 Impact of considering power losses on real instances

We performed a number of what-if analyses on different real-world wind farms. In particular, we were interested in evaluating the impact of considering power losses in the design phase. We will next compare solutions optimized only for CAPEX costs, with solutions optimized looking at the whole lifetime of the park. We will then study the usage of different types of cable (with different resistances) in both cases, and the long-term savings compared with the possibly higher investments costs.

7.5.1 Test instances

We tested our model on the real-world instances proposed in Fischetti and Pisinger, 2017b. They consider five different real wind farms in operation in United Kingdom and Denmark, and one new wind farm under construction. These parks are Horns Rev 1, Kentish Flats, Ormonde, Dan Tysk, Thanet and Horns Rev 3.

This dataset includes old and new parks, with different power ratings and different number of turbines installed, and therefore represents a good benchmark for our tests. Each park has one substation with its own maximum number of connections (C).

In details:

- Horns Rev 1 has 80 turbines Vestas 80-2MW and $C = 10$.
- Kentish Flats has 30 turbines Vestas 90-3MW. It is a near-shore wind farm, so it is connected to the onshore electrical grid without any offshore substation. Nevertheless, only one export cable is connected to the shore, therefore the starting point of the export cable is treated as a substation. We set $C = \infty$ as there is no physical substation limitation in this case.
- Ormonde has 30 Senvion 5MW and $C = 4$.
- DanTysk has 80 Siemens 3.6MW and $C = 10$.
- Thanet has 100 Vestas 90-3MW and $C = 10$.
- Horns Rev 3 has 50 Vestas 164-8MW and $C = 12$ (this is a preliminary layout for this park).

The dataset also includes different sets of cables, indicated as cb01, cb02, cb03, cb04 and cb05.

The cost of the cables considering power losses has been precomputed following the strategy described in Subsection 7.2.2. We computed the cable-loss prices using real data (for Horns Rev 1 and 3, Ormonde and DanTysk) and estimates based on Weibull distributions (Kantish Flats and Thanet).

Each combination of site (i.e., wind farm) and feasible cable set represents an instance in the testbed.

7.5.2 Impact of considering power losses

The aim of this subsection is to analyse how cable routing changes when cable losses are taken into account. We used the real-world instances presented in the previous subsection to perform our tests. We ran our optimization tool with a time-limit of 10 hours (on an Intel Xeon CPU X5550 at 2.67GHz, using Cplex 12.6 as MILP solver) in order to have high quality solutions (for small instances, these are in fact proven optimal solutions).

In all our instances, thicker cables are more expensive and have lower resistance. This means that if the designer of the cable routing aims only at minimizing the

initial costs (CAPEX), then he/she would go for the cheapest cables satisfying the load, thus increasing the power losses. On the contrary, focusing only on minimizing the losses, one would go for the most expensive cables, thus increasing the initial costs. Using the methods explained in Section 7.2.2, we aim at finding the optimal balance between the two objectives, looking at the overall costs in the life time of the park.

As it can be seen from Table 7.4, the amount of savings varies from instance to instance, depending on the prices, on the restrictions of the specific wind farm, and on the structure of the layout.

Table 7.4: Increase in the initial investment and long term savings for our test instances (Net Present Value). The first two columns denote the wind farm and possible cable types. The next column shows how much the investment is increased in the layout taking cable losses into account. In all test cases this amount is paid back in 25 years, and the additional savings by using the lifetime-optimized cable layout are shown in the last column Fischetti and Pisinger, 2017a.

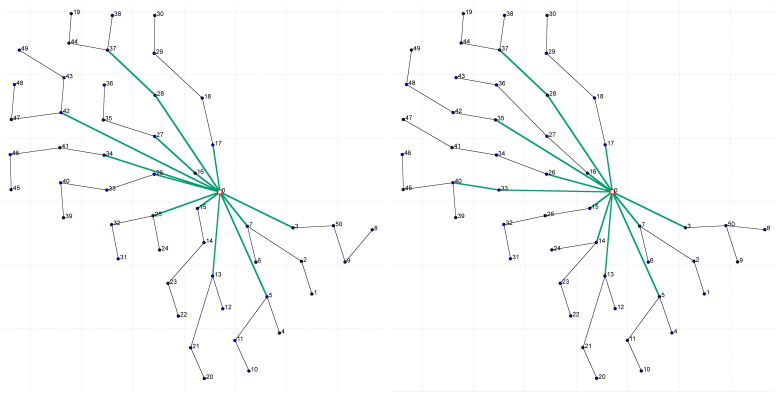
wind farm	cable set	increase in initial investment [K€]	net savings in 25y [K€]
Horns Rev 1	cb01	1	23
	cb02	24	60
	cb05	103	56
Kentish Flats	cb01	2	3
	cb02	1	4
	cb04	19	8
	cb05	5	1
Ormonde	cb03	9	0
	cb04	19	16
DanTysk	cb01	115	21
Thanet	cb04	15	92
	cb05	1	19
Horns Rev 3	cb04	42	172
	cb05	682	208

It should be noticed that the layout optimized on the wind/farm lifetime always provides some savings in the long term, but the amount highly varies from case to case. In Figure 7.5 the case of Horns Rev 3 with cable set cb04 is shown.[1] As expected, the usage of thicker cables (green in the figure) increases in the loss-optimized layout.

In this case the loss-optimized layout is 42 K€ more expensive at construction time (with respect to the CAPEX optimized layout). Nevertheless, in 25 years,

[1] This is a preliminary layout from Vattenfall, not necessarily reflecting the final layout.

Figure 7.5: Optimized cable routing for Horns Rev 3, using cable set cb04. The experts imposed the additional constraint that cable type 2 can support 5 turbines only twice. The first layout is optimized only on CAPEX, the second considers power losses as well Fischetti and Pisinger, 2017a.



this amount is paid back and 172 K€ are additionally saved.

We now try to investigate how the optimizer is restructuring the layout in order to achieve savings in the long run. As already noticed, every wind farm is different, so one cannot define a rule of thumb to design a good cable routing. Nevertheless, observing our layouts, we noticed a different proportion in the usage of the cable types (black and green in the figures). In particular, all the CAPEX solutions minimize the use of the expensive cables: looking only at the immediate costs, it is always preferable to go for the cheapest cable when possible, even creating longer connections. When optimizing considering losses, instead, cables with less resistance become more appealing, even if they are more expensive. In the Horns Rev 1 instance, for example, going from CAPEX optimized to lifetime-optimized the usage of type 1 cables decreases (from 55.5% of the total length to 40.3%) and the usage of type 2 cables increases (from 44.5 to 59.7%).

In Table 7.5 we report the cable usage (percentage of the total cable length) for all our test-bed solutions.

All in all, it can be observed from our results on real-world instances that in most cases it is convenient to invest in cables with lower resistance. The cable route and the type of cable selection for each connection is not an obvious choice and an optimization tool is necessary to determine it.

Table 7.5: Analysis on the usage of different types of cables when optimizing considering or not losses. The last three columns report the usage of the different cable types as percentage of the total cable length of that layout Fischetti and Pisinger, 2017a.

ID	wind farm	cable set	opt mode	% length		
				Type 1	Type 2	Type 3
1	Horns Rev 1	cb01	capex	55.1	40.1	4.8
2			lifetime	53.6	41.7	4.7
3		cb02	capex	57.4	42.6	
4			lifetime	44.1	55.9	
5		cb05	capex	100.0	0.0	
6			lifetime	87.7	12.3	
7	Kentish Flats	cb01	capex	66.4	33.6	0.0
8			lifetime	66.1	33.9	0.0
9		cb02	capex	66.4	33.6	
10			lifetime	60.8	39.2	
12		cb04	capex	90.1	9.9	
13			lifetime	90.1	9.9	
14		cb05	capex	95.6	4.4	
15			lifetime	95.6	4.4	
16	Ormonde	cb03	capex	69.6	30.4	
17			lifetime	76.7	23.3	
18		cb04	capex	66.9	33.1	
19			lifetime	67.4	32.6	
20	DanTysk	cb01	capex	39.0	19.4	41.7
21			lifetime	38.7	22.5	38.8
26	Thanet	cb04	capex	86.3	13.7	
27			lifetime	82.7	17.3	
28		cb05	capex	71.9	28.1	
29			lifetime	71.9	28.1	
30	Horns Rev 3	cb04	capex	57.4	42.6	
31			lifetime	60.7	39.3	
32		cb05	capex	51.8	48.2	
33			lifetime	52.6	47.4	

7.6 Analysis on the energy price K

In this section we will focus on the value K appearing in equation (7.17). As we have seen, K is a factor to estimate the value (in €) of a MW of loss, and is computed as $K = K_{euro} \cdot 8760$ where K_{euro} is the NPV for a MW/h production over the park lifetime, and 8760 is the number of hours in a year. Note that K acts as a balancing factor between the immediate costs (CAPEX) and the power losses. In this section we will investigate the impact of this balancing factor on the final layout. In Subsection 7.6.1, we will perform a multi-criteria analysis where we consider different values of K_{euro}, supposing that the company requests that the extra investment must be paid off in a

limited number of years. Secondly, in Subsection 7.6.2, we will evaluate the impact of considering fluctuating prices depending on the wind scenario.

7.6.1 Sensitivity tests on the return of investment

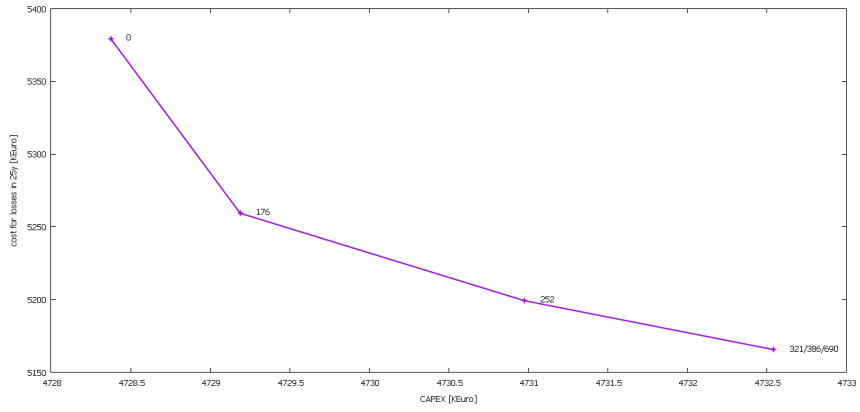
As discussed in Subsection 7.2.2, one has to balance between two opposite objectives: minimizing immediate costs and minimizing revenue losses in the long run. As we have seen in the previous tests, these two objectives are not always aligned since the more expensive cables have lower resistances (so less losses). The balancing factor between the two objectives is K_{euro}, that represents the price of energy (Net Present Value). Setting K_{euro} to zero, for example, means that there is no revenue from selling energy, therefore it does not matter to have losses, but it is instead important only to minimize immediate costs. This corresponds to the case that we called "CAPEX optimized" in the previous tests. On the contrary, setting K_{euro} to a high value, implies that big revenue can be earned selling more energy, so it is very important to minimize losses (whatever initial costs this could imply). The balance between the two objectives, in practice, is set by defining the parameter K_{euro} for the specific project of interest. This is a value known by the designer, and varies from project to project. A realistic value for K_{euro} has been used in the tests of the previous subsection (this value considers weighted average cost of capital (WACC), subsidies for 10 years of operations and estimated market price). Nevertheless, one could be interested in studying how the balance between immediate costs and long term costs varies when varying K_{euro}. As a practical example, one could be interested in optimizing CAPEX and losses at the same time, but being sure to pay off the extra investment in a short time. We considered, in this test, Horns Rev 3 with cable set cb04. For $K_{euro} = 0$ we have our CAPEX solution of Figure 7.5 (top), for $K_{euro} = 690$ €/MWh we have our life-time loss-optimized solution of Figure 7.5 (bottom). Company experts estimated 690 €/MWh to be a realistic value for the energy earning over 25 years of operation (expected lifetime of a wind park). We asked them to recompute this value assuming that we want a return of investment in a shorter time. They recomputed it to be $K_{euro} = 176$ for two years, $K_{euro} = 252$ for 3 years, $K_{euro} = 321$ for 4 years, and $K_{euro} = 386$ for 5 years. Setting our balancing factor K_{euro} to these values translates in imposing that extra CAPEX cost will be paid back in 2, 3, 4 or 5 years, respectively. We recomputed the cable costs according to these different values of K_{euro} and re-optimized the layout accordingly. Once the optimized layouts were found, we re-evaluated them with $K_{euro} = 0$ to evaluate their CAPEX costs and $K_{euro} = 690$ to estimate their cost in 25 years. Table 7.6 shows these figures. For K_{euro} higher than 321 €/MWh the layout is not changing. This means that in the lifetime optimized solution ($K_{euro} = 690$) all

the additional CAPEX costs were actually paid back in 4 years of operation. In Figure 7.6 we plot the values from Table 7.6: the value of the different layouts is decomposed into its CAPEX (x axis) and lifetime-cost part (y axis). The first point (marked by "+" on the leftmost extreme) represents the value for the CAPEX optimized solution ($K_{euro} = 0$): it has the lowest immediate cost, but the highest cost on the long run. Proceeding from left to right, the next "+"s represent the solutions optimized over 2, 3, 4 and 5 years respectively. As already mentioned, from the 4th year on, the layout is not changing any more, and is equal to the solution optimized on the park lifetime ($K_{euro} = 690$), therefore all these layouts are represented at the same coordinates in the plot in Figure 7.6.

Table 7.6: Bi-objective analysis for Horns Rev 3 with cable set cb04: solutions change when varying parameter K_{euro} Fischetti and Pisinger, 2017a.

K_{euro}	immediate cost [k€]	total lifetime cost [k€]	revenue loss due to power losses [k€]
0	47283	52663	5379
176	47291	52551	5259
252	47309	52508	5199
321	47325	52490	5165
386	47325	52490	5165
690	47325	52490	5165

Figure 7.6: Bi-objective analysis from Table 7.6. Each "+" corresponds to a layout optimized for a given value of K_{euro} (specified beside each "+") and its coordinates correspond to its immediate cost (x axis) and costs in 25years (y axis). The layouts optimized with $K_{euro} = 321$, 386, and 690 are the same Fischetti and Pisinger, 2017a.

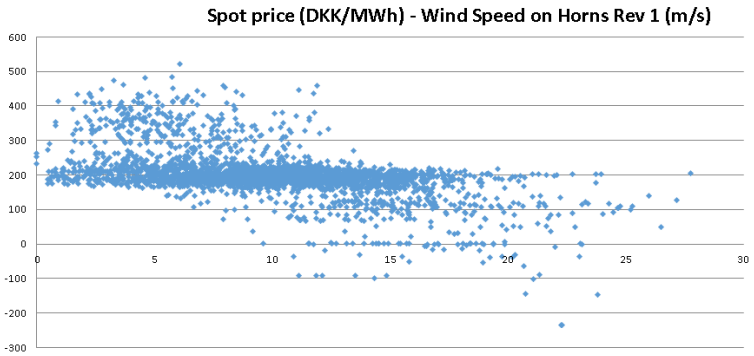


7.6.2 Considerations on price fluctuations

In all our analyses we assumed to have a unique price for energy, independently of the wind scenario. This is in general true, since, at least in Denmark, parks operate with a protected price for about 10 years. Nevertheless, in other countries, this could not be the case, and the price of energy would depend on the market.

In this subsection we suppose not to have a warranted price for wind energy, but to sell energy at the market price. Of course this analysis requires a sufficient amount of data on the spot market price variations. We recorded the Nord Pool prices over the first semester of 2015, sampling the market price every hour together with the wind speed in the park at that time. Figure 7.7 plots these samples against the wind speed in Horns Rev 1.

Figure 7.7: Spot price (DKK/MWh) on the y-axis vs wind speed (m/s) on the x-axis for HR1. Each dot represents a real-world sample recorded in 2015.



It can easily be observed from Figure 7.7 that there is a correlation between energy prices and wind: when there is low wind (under 5 m/s) the price tends to be higher, while when the wind is high (over 12-15 m/s) the price drops. This is because of a surplus of MWh production at high wind speeds.

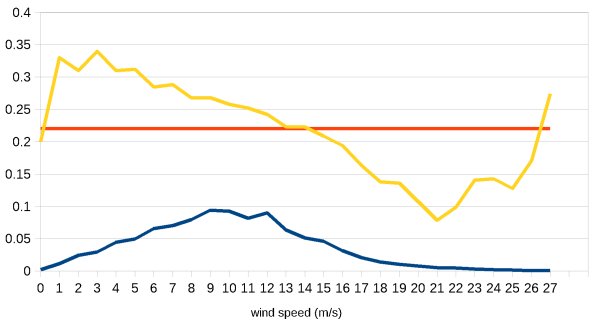
Looking at this analysis, one could then re-consider the power losses figures we have used so far, and investigate the impact of price fluctuations on the cable routing. In order to do so, we had to reformulate the loss cost-related part of the objective function, considering that now the value K_{euro} depends itself on

the wind scenario s, and therefore will be indicated as K^s_{euro} in the following. The value $loss^{tf}$ to precompute is now

$$loss^{tf} = 3R^t \sum_{s \in S} \pi_s (fI^s)^2 K^s_{euro}. \tag{7.20}$$

We consider again the Horns Rev 1 case in our test, with the same cable set as in Section 7.4. In order to estimate the impact of considering a K^s_{euro} that varies with scenarios s, we compared with the case of a fixed K_{euro}, equal to the average spot price. In both cases we considered a WACC of 8%. Figure 7.8 shows the two options: in yellow, the value of K_{euro} that varies over the different wind speeds (x-axis); in red, the value of K_{euro} that is fixed at the average market price (0.22 €/KWh). The value of K_{euro} in the varying case (yellow line) has been computed by a simple interpolation of the registered spot prices (Figure 7.7), by computing their average at each wind speed. Note that, in formula (7.20), the different scenarios s are weighted by their probability π_s: the blue line in Figure 7.8 represents the probabilities used in our test case (extracted from samples of real-data from Horns Rev).

Figure 7.8: K_{euro} variations over different wind speeds in the two possible approaches: considering one fixed value equal to the average market price (red line), or explicitly considering the price variations (yellow line). The blue line shows the frequency of the different wind scenarios in the site.



We computed the cable prices for the two strategies, using the precomputing strategy of Subsection 7.2.3 with formula (7.17) for fixed price, or with formula (7.20) for fluctuating price. Table 7.7 shows the result of the precomputation: the first two columns give the details of the cable set (type of cables, and capacities in terms of number of turbines), the third column reports CAPEX prices, the fourth column the prices computed considering losses with a fixed

K_{euro}; and the fifth column considering a price of energy that varies over different wind scenarios.

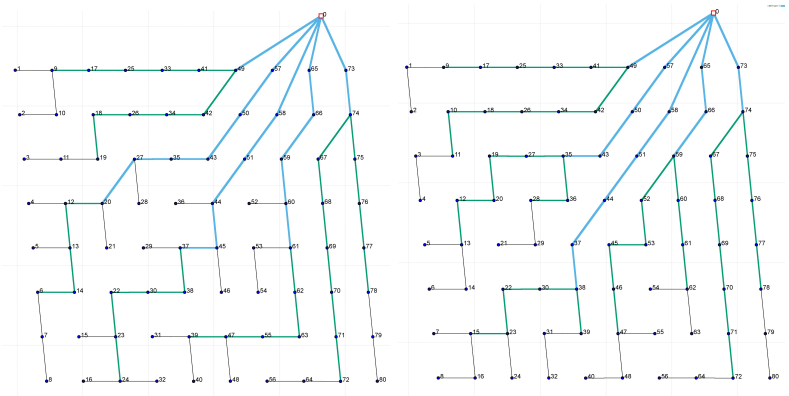
Table 7.7: Costs for the cables. In the first two columns we indicate the type of cable and its capacity, expressed as number of 2MW turbines supported. The third column reports the CAPEX costs of the cables, including installation costs. The last columns show the cable prices taking power losses into account: in the fourth column assuming a fixed price of energy (i.e., the average market price), in the fifth column considering a different price of energy for each scenario.

type	n.turb.	capex	avg.	fluctuating
1	1	345	345.56	345.56
2	2	385	385.85	385.85
	3	385	386.65	386.64
	4	385	387.77	387.76
	5	385	389.21	389.19
	6	385	390.97	390.95
	7	385	393.04	393.02
	8	385	395.44	395.41
3	9	500	509.12	509.09
	10	500	511.21	511.17
	11	500	513.52	513.47
	12	500	516.05	515.99
	13	500	518.80	518.74
	14	500	521.77	521.69
	15	500	524.96	524.87
	16	500	528.37	528.27

As already discussed, until the max capacity of the cable is reached, the CAPEX cost (third column) does not depend on the number of turbines connected – while the costs including losses (fouth-fifth column) do. Comparing the last two columns, it can be noticed that the input cable prices are not very sensitive to the variation of K_{euro}. This is also explained by the fact that the extreme wind speeds, where K_{euro} varies the most, are also the less frequent ones. At the most frequent wind speeds (between 5 and 15 m/s) K_{euro} is closer to its average value (see Figure 7.8). If we run the optimization tool on the HR1 case using the losses prices, we obtain the two layouts in Figure 7.9. As in our previous layout plots, different colours indicate different cable types. Referring to Table 7.7, black lines represent cable type 1, green cable type 2, and blue cable type 3.

Assume that the company decides to use the layout optimized for the average price (first plot in Figure 7.9). If we re-evaluate this layout considering fluctuating prices, we can conclude that the company would loose 1400 € (0.006%)

Figure 7.9: Layout considering no protected price of energy: the first plot shows the optimized layout considering a fixed price of energy, equal to the average market price. The second plot shows the optimized layout considering price variations over the different wind scenarios.



in the park lifetime, with respect with the fluctuating-price layout (second plot in Figure 7.9). This shows that the impact of considering fluctuating prices is, in this example, very small.

All in all, our results suggests that price fluctuations do not significantly impact the layout. Even if it is less profitable to avoid losses when the price of energy drops, this event is so rare in reality that it does not pay off to consider it in the cable routing optimization.

7.7 Conclusions

In this paper we used Mixed Integer Linear Programming (MILP) and Matheuristic approaches to optimize inter-array offshore cable routing. The main focus of the paper is to quantify the impact of considering both the immediate cable costs and power losses already at design phase. First, we illustrated how to mathematically model the problem and how to deal with large scale instances using a matheuritic approach. Next, we performed different analyses on real-world instances. To begin with, we compared the optimized solution with an existing cable layout, proving that more than one million Euro can be saved by using adequate optimization tools for the offshore cable routing prob-

lem. Afterwords, we compared optimized solutions under different assumptions to understand and (for the first time) quantify the impact of considering cable losses in real offshore cable routings.

In general, we observed that it is convenient to use cables with less resistance in order to reduce power losses, even if these cables are more expensive at construction time. We used our testbed to evaluate the profitability of the new solutions, both in terms of CAPEX and revenue in the long term. Results show that it is very difficult to define some "rules-of-thumb" for this problem, since usage of cables and savings highly vary from instance to instance. This proves that a proper optimization tool, as the one presented here, is necessary for an optimal design of each layout. Finally, we performed different analyses on the balancing parameter K_{euro}. This corresponds to giving more or less importance to power losses in the objective function, and it is of great importance for designers. In this way, indeed, they can evaluate the return of investment and the impact of their assumptions on the long-term energy price, when designing their cable routing. In particular, we looked at two specific reasons for which the company could consider different energy prices: requirements on the return of investments and fluctuations of the energy price on the market. In the latter case, we extended the original model to consider the dependency of the energy price over the different wind scenarios, using real-world measurements. Our tests showed that it is important to define a value of K_{euro} that well reflects the requirements of the specific project, whereas the layout is not very sensitive to small variations of this parameter.

References

Bauer, J. and J. Lysgaard (2015). "The offshore wind farm array cable layout problem: a planar open vehicle routing problem". In: *Journal of the Operational Research Society* 66.3, pp. 360–368.

Berzan, C., K. Veeramachaneni, J. McDermott, and U. O. Reilly (2011). "Algorithms for cable network design on large-scale wind farms". In: *Tech. Rep. Tufts University*.

Boschetti, M., V. Maniezzo, M. Roffilli, and A. R. Bolufé (2009). "Matheuristics: Optimization, Simulation and Control". In: *Hybrid Metaheuristics*. Springer Berlin Heidelberg, pp. 171–177.

Cerveira, A., A. D. Sousa, E. J. S. Pires, and J. Baptista (2016). "Optimal Cable Design of Wind Farms: The Infrastructure and Losses Cost Minimization Case". In: *IEEE Transactions on Power Systems* 31.6, pp. 4319–4329.

Dutta, S. (2012). "Data Mining and Graph Theory Focused Solutions to Smart Grid Challenges". MA thesis. University of Illinois.

Dutta, S. and T. J. Overbye (2011). "A Clustering Based Wind Farm Collector System Cable Layout Design". In: *Power and Energy Conference at Illinois (PECI)*, pp. 1–6.

Fagerfjall, P. (2010). "Optimizing Wind Farm Layout - More Bang for the Buck Using Mixed Integer Linear Programming". MA thesis. Goteborg, Sweden: Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University.

Fischetti, M. and D. Pisinger (2017a). "On the impact of using Mixed Integer Programming techniques on real-world offshore wind parks". In: *Proceedings of the 6th International Conference on Operations Research and Enterprise Systems*.

Fischetti, M. and M. Fischetti (2016). "Matheuristics". In: *Handbook of Heuristics*. Ed. by R. Martí, P. Panos, and M. G. Resende. Springer International Publishing, pp. 1–33. ISBN: 978-3-319-07153-4. DOI: 10.1007/978-3-319-07153-4_14-1. URL: http://dx.doi.org/10.1007/978-3-319-07153-4_14-1.

Fischetti, M. and D. Pisinger (2017b). "Optimizing wind farm cable routing considering power losses". In: *European Journal of Operational Research*. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.07.061. URL: http://www.sciencedirect.com/science/article/pii/S037722171730704X.

González, J. S., M. B. Payán, J. M. R. Santos, and F. González-Longatt (2014). "A review and recent developments in the optimal wind-turbine micro-siting problem". In: *Renewable and Sustainable Energy Reviews* 30, pp. 133–144.

González-Longatt, F. M. and P. Wall (2012). "Optimal Electric Network Design for a Large Offshore Wind Farm Based on a Modified Genetic Algorithm Approach". In: *IEEE Systems Journal* 6.1, pp. 164–172.

Hansen, P., V. Maniezzo, and S. Voß (2009). "Special issue on mathematical contributions to metaheuristics editorial". In: *Journal of Heuristics* 15.3, pp. 197–199.

Hertz, A., O. Marcotte, A. Mdimagh, M. Carreau, and F. Welt (2012). "Optimizing the Design of a Wind Farm Collection Network". In: *INFOR* 50.2, pp. 95–104.

Kristoffersen, J. and P. Christiansen (2003). "Horns Rev offshore windfarm: Its main controller and remote control system". In: *Wind Engineering* 27.5, pp. 351–359.

Li, D., C. He, and Y. Fu (2008). "Optimization of Internal Electric Connection System of Large Offshore Wind Farm with Hybrid Genetic and Immune Algorithm". In: *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT2008)*, pp. 2476–2481.

Pillai, A., J. Chick, L. Johanning, and M. K. V. D. Laleu (2015). "Offshore wind farm electrical cable layout optimization". In: *Engineering Optimization* 47.12, pp. 1689–1708.

Zhao, M., Z. Chen, and F. Blaabjerg (2009). "Optimisation of Electrical System for Offshore Wind Farms via Genetic Algorithm". In: *IET Renewable Power Generation 3* 3.2, pp. 205–216.

Chapter 8

Optimal wind farm cable routing: modeling branches and offshore transformer modules

Martina Fischetti[a] · David Pisinger[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]Management Science, Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

Publication Status: Accepted in : Fischetti and Pisinger, 2018

Reading Instructions: This paper extends the model of Fischetti and Pisinger, 2017 to include new technical constraints and evaluate new technologies on the market. If you read Chapter 6, you can skip Section 8.2

Abstract: Many EU countries aim at reducing fossil fuels in the near future, hence an efficient production of green energy is very important to reach this goal. In this paper we address the optimization of cable connections between turbines in an offshore wind park. Different versions of the problem have been studied in the recent literature. As turbines are becoming still more customized, it is important to be able to evaluate the impact of new technologies with a flexible optimization tool for scenario evaluation. In a previous joint project with Vattenfall BA Wind (a global leader in energy production) we have studied and modelled the main constraints arising in practical cases. Building on that model, in the present paper, we address new technological features that have been recently proposed by Vattenfall's experts. We show how some new features can be modelled and solved using a Mixed-Integer Linear Programming paradigm. We report and discuss computational results on the performance of our new models on a set of real-world instances provided by Vattenfall.

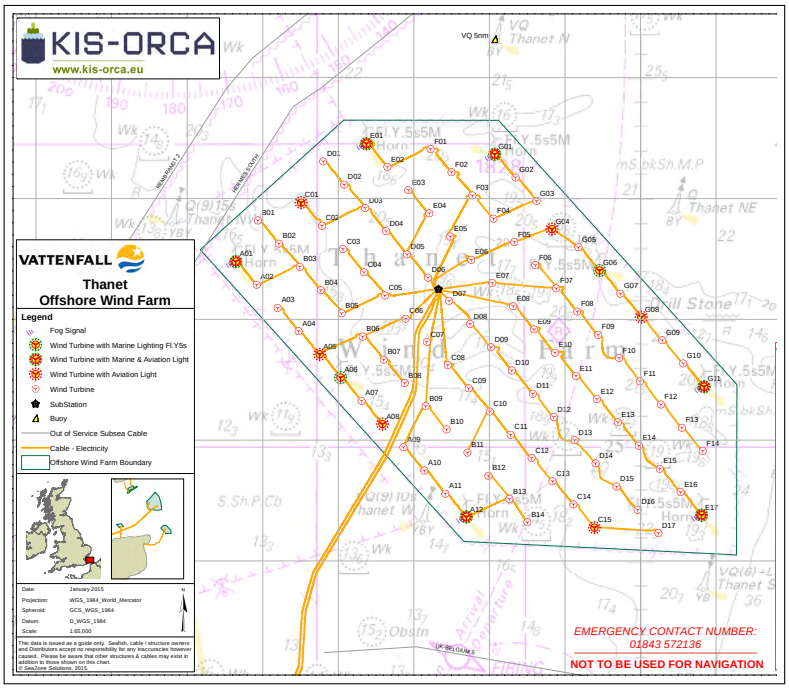
8.1 Introduction

The production of green (in particular, wind) energy is an important instrument in limiting the climate changes of the world. As modern wind parks are getting bigger in size and in produced power, it is very important to optimize their design. Designing a wind farm is, however, a complex process including selection of the right site, optimizing the location of each turbine Fischetti and Monaci, 2015; González et al., 2014, establishing the infrastructure Bauer and Lysgaard, 2015 and connecting the farm to the existing electrical grid Qi et al., 2015.

In this paper we address the optimization of cable connections among offshore turbines, called *cable routing* in what follows. When turbines are located offshore their energy production is first transmitted to one or more collection points (*substations*) using lower voltage cables (33 or 66 kV), called *inter-array cables*. The energy is then moved from the substations to shore using higher voltage cables, called *export cables*. We will here focus on the inter-array cable

optimization. The original version of this problem consists of connecting all the offshore turbines to one (or more) offshore substation(s), minimizing the total cable cost. The final cable layout has a tree structure where the non-root nodes correspond to the given turbines, the substations play the role of roots, and the energy (i.e., the electric current) flows from the nodes to the roots along the tree. Figure 8.1 gives an example of cable layout for a real wind park.

Figure 8.1: An example of cable routing for a real-world offshore wind park (Thanet) owned by Vattenfall—picture from Kis-orca, 2015.



A number of constraints must be taken into account when designing a feasible cable routing. First of all, the energy flow is unsplittable, i.e., the flow leaving a turbine must be supported by a single cable. In addition, each substation has a physical layout that imposes a maximum number of entering cables. Cable crossings should be avoided, as establishing one cable across another is expensive and increases the risk of cable damages.

Several types of cables with different costs and capacities are available on the market. Therefore, one has to also optimize the cable type selection in order to deliver all the energy production to the substations at minimum cost. In our collaboration with Vattenfall, we had the chance to have a close look at

how engineers are evaluating different scenarios and technological possibilities, to design competitive wind parks. Earlier, most of this work was carried out manually, so evaluating different possibilities was very difficult and time consuming. We closely collaborated with different engineer teams in Vattenfall, to model and optimize different versions of the cable routing problem arising in practice. This family of problems has received limited attention in the OR community so far, so we aim here at describing and modelling some new optimization problems from an OR perspective, while also showing the impact of having sound optimizers to help engineers in practice. To be more specific, we will first describe different versions of the classical cable routing problem arising in practical applications, and then we will compare the resulting layouts following the "what-if" analysis approach that is carried out by the company before selecting one technology instead of another.

The basic formulation of the wind park cable routing optimization problem has received significant attention in the OR literature in the last years. Due to the large number of constraints and the intrinsic complexity of the problem, many papers (including Dutta and Overbye, 2011; González-Longatt and Wall, 2012; Li et al., 2008; Pillai et al., 2015; Zhao et al., 2009 among others) prefer to use ad-hoc heuristics. Just few articles from the literature use Mixed Integer Linear Programming (MILP) for cable routing; see e.g. Bauer and Lysgaard, 2015; Berzan et al., 2011; Cerveira et al., 2016; Dutta, 2012; Fagerfjall, 2010; Hertz et al., 2012.

Compared to Bauer's definition Bauer and Lysgaard, 2015 of the Offshore Wind Farm Array Cable Layout (OWFACL) problem, our initial formulation also includes substation limitations and the possibility of having different cable types. Therefore, we decided to introduce a new name for our version of the problem, denoted the Offshore Wind Farm Cable Routing (OWFCR) problem. The new formulation allows for multiple substations as well.

The basic formulation of OWFCR has been studied by the present authors in Fischetti and Pisinger, 2017, considering also additional technical features such as obstacles in the site and power losses in the cables.

Due to our ongoing collaboration with Vattenfall, we have a continuous feedback from experts on the new problem specifications arising from upcoming projects. Wind energy is a highly-competitive and a relatively new field, where technology is still quickly developing. Suitable MILP models that can capture new technology requirements are therefore very valuable in scenario evaluation as a substitute for, or a complement to, the existing manual design process. In this paper we will therefore look at different possible scenarios for the ca-

ble routing problem, where different technological requirements are considered. We will see how the basic MILP model (OWFCR) can be extended to evaluate these scenarios and be used to quantify their impact on the design of offshore wind parks. In particular, we will present four different extensions of the original OWFCR model, that will be discussed in details in the next sections.

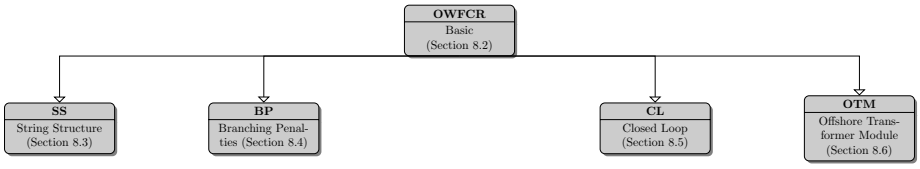
The paper is organized as follows. In Section 8.2 we recall the original version of the cable routing problem (OWFCR) and explain how to model its basic constraints using Mixed Integer Linear Programming. When discussing the resulting layouts with practitioners, we received different additional requirements that we included in the new versions of the model.

First, we considered an engineering requirement on the topology of the route: we were asked to evaluate the impact of connecting turbines only in strings, i.e., to have at most one cable entering and exiting each turbine. The resulting model will be denoted as OWFCR-SS (String Structure), and will be studied in Section 8.3. Secondly we were asked to not impose any specific topology, but explicitly consider the possible additional costs for connecting more cables to each turbine (as additional switch-gears may be needed in some turbine models). This version of the model is denoted OWFCR-BP (as it considers Branching Penalties). It is studied in Section 8.4.

Even though we could show that a layout with branches is less expensive from an immediate cost perspective, it could be difficult for the company to detect cable failures in such a structure. Therefore, the company proposed an additional scenario, where turbines must be connected in strings and the strings must be coupled in loops to create fault-tolerant cycles. The problem of constructing fault-tolerant networks is well know in the telecommunication literature, where different topologies have been studied (see, e.g. Carroll et al., 2013; Fortz and Labbé, 2004; Labbé et al., 2004). Nevertheless, in the considered wind farm application, we were asked by the company to consider a very specific topology, namely, a so-called *closed-loop structure*. Since the aim of our work is to answer the practical needs of our industrial partner, we decided to stick to this structure. A closed-loop (or ring) structure is characterized by the presence of redundant minimum-capacity cables between strings, that avoid disconnected turbines in case of cable failure. We analyze this structure in Section 8.5, where we extend our model to handle ring structures, resulting in our third variant of the model (OWFCR-CL). As we will see also in our computational section, this is a more expensive structure (compared with the original branch structure) so it should be used only if necessary (i.e., if turbines are not equipped to survive cable failures by other means).

In Section 8.6 we consider the extension of the OWFCR model where there is no offshore substation, but a smaller Offshore Transformer Module (OTM) can be installed on normal turbines. The turbines equipped with this OTM can be connected to both inter-array cables (small cables that connect turbines one with each other) and to the export cable (higher voltage cable, that connects turbines to shore). The new variant of the problem, named OWFCR-OTM, is a particularly interesting problem variant, since it involves also the decision of how many OTMs should be installed and on which turbines. Given the high cost of offshore substations, a properly optimized layout with OTMs can greatly reduce the overall costs. Each of the new models extends the original OWFCR model, as shown in the diagram in Figure 8.2.

Figure 8.2: Taxonomy of the OWFCR problem variants



Due to the complexity and size of the studied problems, we used matheuristic techniques to speed-up the solution of the previously described models. Section 8.7 briefly describes the hybrid matheuristic/exact algorithm we used. Section 8.8 reports our tests of the new models on a set of real-world instances, while some conclusions are drawn in Section 8.9.

8.2 The basic MILP model (OWFCR)

We first need to briefly recall the basic model (OWFCR) we developed for Vattenfall; the reader is refereed to Fischetti and Pisinger, 2017 for further details.

Assuming that turbine positions are fixed and a set of cable options (with different capacities and costs) is given, the OWFCR problem is to find an optimal cable connection between all turbines and the given substation(s), minimizing the total cable cost. The network must ensure that the energy flow on each link does not exceed the capacity of the installed cable, and the energy flow leaving

a turbine is supported by a single cable. An additional technical requirement is that a given maximum number of cables, say C, can be connected to each substation. Finally, cable crossings should be avoided to reduce the risk of damages.

8.2.1 Mathematical formulation

Turbines can be represented by nodes in a complete and loop-free directed graph $G = (V, A)$, and all possible connections between them by directed arcs. Some nodes correspond to the substations that are considered as the roots of the distribution network, and are the only nodes that collect energy. The final solution consists of a set of trees rooted at the substations whose arcs are directed from the nodes to the roots, following the energy flow. The model also allows for optional "Steiner" nodes, that can either be left uncovered, or have exactly one entering and one leaving cable. These dummy nodes are useful when considering obstacles in the area, or to allow for curvy connections between two nodes; see Fischetti and Pisinger, 2017 for details.

Each node corresponds to a point in the plane, whose coordinates are used to compute distances between nodes as well as to determine whether two given line segments $[i, j]$ and $[h, k]$ cross each other, where $[a, b]$ denotes the line segment in the plane having nodes $a, b \in V$ as endpoints. It is assumed that two line segments meeting at one extreme point do not cross each other. Analogously, two segments do not cross if one is contained in the other, as they correspond to two parallel cables that can be physically built one besides the other without crossing issues.

The node set V is partitioned into (V_T, V_0, V_S), where V_T contains the nodes corresponding to the turbines, V_0 contains the nodes corresponding to the substations, and V_S contains the Steiner nodes (if any). Furthermore, let $P_h \geq 0$ denote the power production at node $h \in V$, where $P_h > 0$ for $h \in V_T$ and $P_h = 0$ for $h \in V_S$ (nodes $h \in V_0$ corresponding to substations have $P_h = -1$ by convention).

Let T denote the set of different cable types that can be used. Each cable type $t \in T$ has a given capacity $k_t \geq 0$ and a unit cost $u_t \geq 0$. Arc costs $c^t_{i,j} = u_t \cdot dist(i, j)$ can then be computed for each arc $(i, j) \in A$ and for each cable type $t \in T$, where $dist(i, j)$ is the Euclidean distance between nodes i and j.

Decision variables are as follows. For each arc $(i, j) \in A$, we have a continuous

variable $f_{i,j} \geq 0$ representing the (directed) energy flow from i to j, and a binary variable $x^t_{i,j} = 1$ iff arc (i,j) is constructed with cable type $t \in T$. Finally, binary variables $y_{i,j} = \sum_{t \in T} x^t_{ij}$ indicate whether an arc (i,j) is built with any type of cable.

The MILP model presented in Fischetti and Pisinger, 2017 will be denoted OWFCR, and it is defined by:

$$\min \sum_{(i,j) \in A} \sum_{t \in T} c^t_{i,j} x^t_{i,j} \tag{8.1}$$

$$\sum_{t \in T} x^t_{i,j} = y_{i,j}, \quad (i,j) \in A \tag{8.2}$$

$$\sum_{i \in V : i \neq h} (f_{h,i} - f_{i,h}) = P_h, \quad h \in V_T \cup V_S \tag{8.3}$$

$$\sum_{t \in T} k_t \, x^t_{i,j} \geq f_{i,j}, \quad (i,j) \in A \tag{8.4}$$

$$\sum_{j \in V : j \neq h} y_{h,j} = 1, \quad h \in V_T \tag{8.5}$$

$$\sum_{j \in V : j \neq h} y_{h,j} = 0, \quad h \in V_0 \tag{8.6}$$

$$\sum_{j \in V : j \neq h} y_{h,j} \leq 1, \quad h \in V_S \tag{8.7}$$

$$\sum_{i \in V : i \neq h} y_{i,h} \leq 1, \quad h \in V_S \tag{8.8}$$

$$\sum_{i \in V : i \neq h} y_{i,h} \leq C, \quad h \in V_0 \tag{8.9}$$

$$y_{i,j} + y_{j,i} + y_{h,k} + y_{k,h} \leq 1, \quad \text{for all crossing segments } [i,j] \text{ and } [h,k] \tag{8.10}$$

$$x^t_{i,j} \in \{0,1\}, \quad (i,j) \in A, t \in T \tag{8.11}$$

$$y_{i,j} \in \{0,1\}, \quad (i,j) \in A \tag{8.12}$$

$$f_{i,j} \geq 0, \quad (i,j) \in A. \tag{8.13}$$

The objective function (8.1) minimizes the total cable layout cost. Constraints (8.2) say that only one type of cable can be selected for each built arc, and define the $y_{i,j}$ variables. Constraints (8.3) stipulate that the energy (flow) exiting each node h is equal to the energy entering h plus the power production of that node; these constraints are not imposed for $h \in V_0$, i.e., when h corresponds

to a substation. Constraints (8.4) instead ensure that the flow does not exceed the capacity of the installed cable. Constraints (8.5) impose that only one cable leaves a turbine, whereas constraints (8.6) say that no cable can exit a substation, thus enforcing a tree structure rooted at the substation(s). As to Steiner nodes, it is optional to connect them but, if they are connected, only one cable can enter these nodes (constraints (8.7)). Furthermore, (8.8) imposes that at most one cable can exit a Steiner node. Constraint (8.9) imposes the maximum number of cables (C) that can enter each substation. Finally, inequalities (8.10) forbid building any two crossing arcs.

No-cross constraints (8.10) can be strengthened by exploiting constraints (8.5)-(8.7), so as to reduce their number and to improve their quality. To this end, for any node triple (a, b, k), let the "clique" arc subset $\mathcal{Q}(a, b, f)$ be defined

$$\mathcal{Q}(a,b,f) = \{(a,b),(b,a)\} \cup \{(f,h) \in A : \text{ segments } [a,b] \text{ and } [f,h] \text{ cross}\}$$

The following *improved no-cross constraints* have been shown to be valid in Fischetti and Pisinger, 2017:

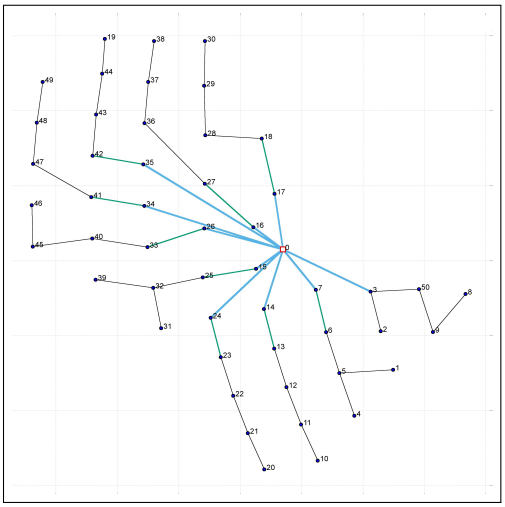
$$\sum_{(i,j)\in\mathcal{Q}(a,b,f)} y_{i,j} \leq 1, \qquad a,b,f \in V,\ a \neq b, b \neq f, f \neq a. \tag{8.14}$$

8.2.2 Example

First we will illustrate the result of our OWFCR model with a real-world example. We consider the Horns Rev 3 (HR3) case, a 350 MW park in Denmark, still under construction. Fifty 8 MW turbines are used in the layout (they are represented as black dots in the figure plots). In this park, the offshore substation is given by the grid operator, thus its position is fixed. At most 12 cables can be connected to the substation (red square in Figures 8.3). Our set of cables consists of three types of cable: the black one supports 3 turbines at a cost of 393 €/m, the green one supports 4 turbines at a cost of 460 €/m, and the blue one supports 5 turbines at a cost of 540 €/m (costs include both cable and installation costs). HR3 will be used as an example park also for the OWFCR model variants, in the next sections. A further comparison between the different models on various real-world wind parks is presented in Section 8.8.

The OWFCR model results in the optimized layout of Figure 8.3 and does not consider other additional costs or constraints. This optimal solution was found in 176 seconds on a standard PC, using the hybrid matheuristic/exact framework of Section 8.7.

Figure 8.3: Optimal solution of the the OWFCR problem for HR3. Power is transmitted from the leaves towards the root (node 0, marked with a red square).



8.3 String structure (OWFCR-SS)

The OWFCR model presented in Section 8.2 often constructs cable routings with branches, as seen in the HR3 example in Figure 8.3. Branches are structures with more than two cables entering a turbine (as node 3, 5, and 32 in Figure 8.3). Branches are not impossible in practice but they can involve extra costs for the additional hardware (load breaker or disconnectors). This depends on the turbines used: some of them are equipped with hardware allowing for multiple connections by default, some of them do not. For turbines with multiple connections the OWFCR problem formulation holds, while for turbines with only one entering cable connection, we were ask to evaluate a scenario where the string structure is enforced. This additional requirement gives rise to the OWFCR-SS problem variant.

8.3.1 Mathematical formulation

The OWFCR model is extended with the following constraints:

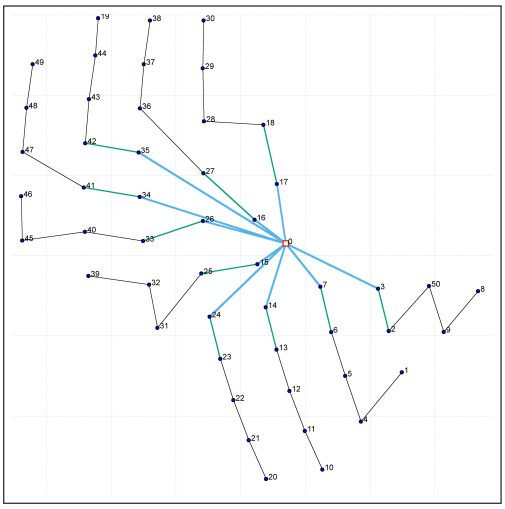
$$\sum_{i \in V : i \neq h} y_{i,h} \quad \leq 1, \quad h \in V_T, \tag{8.15}$$

imposing that at most one cable can enter each turbine. The OWFCR-SS model is then the OWFCR model (8.1)–(8.14) with the addition of constraint (8.15).

8.3.2 Example

As an example of an optimized layout imposing a string structure we again use the HR3 case. We run the OWFCR-SS model on a standard PC with a time limit of 1 hour. The resulting layout is shown in Figure 8.4, and it is 222 k€ more expensive than the one in Figure 8.3. This optimal solution was found in 598 seconds on a standard PC, using the approach described in Section 8.7.

Figure 8.4: Imposing a string structure, i.e., solution of the OWFCR-SS problem.



8.4 Branching Penalties (OWFCR-BP)

Having quantified the cost impact of imposing a string structure on the layout, the company was interested in finding a middle-way solution. Knowing the cost of the additional hardware for branches, and having a tool able to consider this

in the optimization, the company would like to find the the optimal indegree for each turbine in a layout. Note the the extra costs (called *branch penalties* in what follows) depend on the number of arcs entering each turbine, but not in a linear way. We therefore studied another version of the original OWFCR problem, explicitly including branching penalties in the optimization (denoted as the OWFCR-BP).

8.4.1 Mathematical formulation

We describe the modifications needed to deal with branch penalties in the OWFCR model. Let d_{max} be the maximum allowed in-degree for a node (in typical applications, d_{max} is 2 or 3), and let $D = \{1, \cdots, d_{max}\}$. Moreover, for each possible number of entering cables $d \in D$, let π_d be the extra-cost (penalty) incurred for each node in V_T that has in-degree equal to d in the final solution. In our study we considered turbines with at most two entering cables. The standard technology for turbines includes connections for only one entering and leaving cable (this configuration has zero extra cost, so $\pi_1 = 0$). If we want to have more than one entering cable (branch structure), we have to pay for the additional load breakers or disconnectors, and for the extra time to install them. Referring to our test-case HR3, Vattenfall's experts estimated that the extra cost for having two cables entering a turbine is 15 k€for the disconnector, plus 10 k€ for installation (therefore, $\pi_2 = 25$ k€).

We introduce a new set of binary variables z_j^d with $j \in V_T$ and $d \in D$, where $z_j^d = 1$ iff the in-degree of node j is equal to d. The objective function for the OWFCR-BP then reads

$$\min \sum_{(i,j)\in A} \sum_{t\in T} c_{i,j}^t x_{i,j}^t + \sum_{d\in D} \pi_d \sum_{j\in V_T} z_j^d \tag{8.16}$$

while we add the following additional constraints:

$$\sum_{i\in V: i\neq j} y_{i,j} = \sum_{d\in D} d\, z_j^d, \qquad j \in V_T \tag{8.17}$$

$$\sum_{d\in D} z_j^d \leq 1, \qquad j \in V_T \tag{8.18}$$

$$z_j^d \in \{0,1\}, \qquad j \in V_T. \tag{8.19}$$

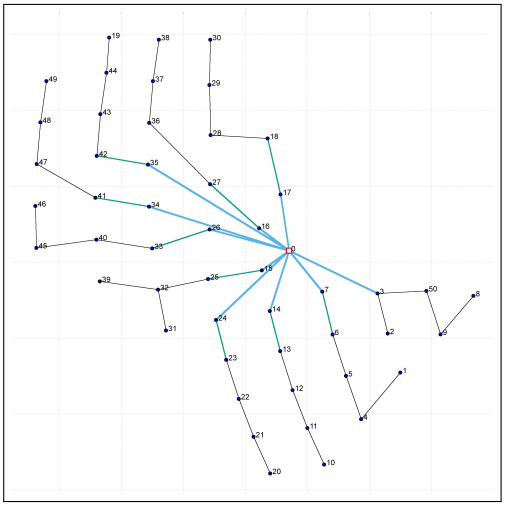
Model OWFCR-BP is therefore the OWFCR model (8.2)–(8.14) with the addition of constraints (8.17)–(8.19), and with (8.16) replacing the objective function.

Note that, in (8.18), we write "$\leq$" instead of "$=$" so as to allow for leaf nodes with zero in-degree.

8.4.2 Example

Figure 8.5 plots the layout obtained by considering the HR3 case with the same constraints as in the OWFCR model, but imposing that at most two cables can enter a turbine and that the extra cost for each of these branches is 25 k€. This optimal solution was found in 232 seconds on a standard PC. The structure of the solution is not really affected by the extra costs, and only slightly changes from Figure 8.3, nevertheless this solution is 20 k€ cheaper than the one in Figure 8.3 and about 200 k€ cheaper than the one in Figure 8.4.

Figure 8.5: Including branch penalties: solution of the OWFCR-BP problem.



8.5 Closed-loop structure (OWFCR-CL)

A main reason to consider the so-called closed-loop structure is to cope with cable failures. Generally speaking, turbines are designed to be connected to an electrical grid. Modern turbines are manufactured to have a certain autonomy to survive disconnection from the grid, but less recent models do not have this

feature. This means that, in case of cable failures, the disconnected turbines could suffer from major damages. In order to avoid this situation, parks with this kind of turbines need to use redundant cables (or expensive batteries/diesel generators attached to each turbine). The main purpose of these redundant cables is to keep all turbines connected to the grid, in case an inter-array cable failure occurs. Note that this extra cable does not need to transport all the produced power to the substation, since turbines are curtailed to reduce their power production in case of cable failure. A specific redundant cable-routing structure was required by our company partner, that we call *closed-loop* (or *ring*) structure. This is the redundant structure most used in practice, because it has high reliability and permits ease of fault location Sannino et al., 2006. This structure consists in having at most one cable entering a turbine (as in the OWFCR-SS formulation) and in pairing the leaf turbines by redundant cable connections; see Figure 8.6 for an illustration. These connections always use only the cheapest cable available, because they are only intended to keep the turbines connected to the grid in case of a cable failure. This new variant of the OWFCR problem with closed loops, will be denoted by OWFCR-CL.

8.5.1 Mathematical formulation

To impose the closed-loop structure in the OWFCR model we introduce a new binary variable $q_{i,j}$ for each $(i,j) \in A$, where $q_{i,j} = 1$ if a redundant cable has to be installed between nodes i and j. These variables are added to the original OWFCR model, so that the new model will find a min-cost set of rings. As the redundant cable connections have no orientation, we actually fix $q_{i,j} = 0$ whenever $i > j$, thus halving the number of additional variables required.

The new variables $q_{i,j}$ are then linked to the $y_{i,j}$ through the following constraints to be added to the basic OWFCR MILP model:

$$\sum_{i \in V: i \neq h} (y_{i,h} + y_{h,i} + q_{i,h} + q_{h,i}) = 2 \sum_{j \in V: j \neq h} y_{h,j}, \qquad h \in V_T \cup V_S \quad (8.20)$$

$$q_{i,j} = 0, \qquad (i,j) \in A : i \geq j \quad (8.21)$$

$$q_{i,j} \in \{0,1\}, \qquad (i,j) \in A. \quad (8.22)$$

Note that the degree-2 constraints (8.20) automatically impose a string structure, with an even number of strings paired into rings. These constraints are not imposed for nodes $h \in V_0$ that correspond to substations (that are allowed to have degree 4 or more), and that the right-hand side term is zero in case node $h \in V_S$ is left uncovered. To avoid that the new arcs induce crossings

in the final routing, in our branch-and-cut solver we dynamically separate (for integer solutions only) the following extended no-cross constraints

$$y_{i,j} + y_{j,i} + y_{h,k} + y_{k,h} + q_{i,j} + q_{j,i} + q_{h,k} + q_{k,h} \leq 1 \tag{8.23}$$

for each pair $[i, j]$ and $[h, k]$ of crossing edges.

As to the objective function, each new variable $q_{i,j}$ has a cost computed as $c_{i,j}^{t_{min}} = u_{min} \cdot dist(i, j)$, where $u_{min} = \min_{t \in T} u_t$ is the unit cost of the least expensive cable. These costs are added to the OWFCR objective function (8.1) to obtain:

$$\min \sum_{(i,j) \in A} \sum_{t \in T} c_{i,j}^t x_{i,j}^t + \sum_{(i,j) \in A} c_{i,j}^{t_{min}} q_{i,j} \tag{8.24}$$

Hence, the OWFCR-CL model is the OWFCR model (8.2)–(8.14) with the addition of constraints (8.20)–(8.23), and the amended objective function (8.24).

8.5.2 Example

Figure 8.6: Closed-loop structure (formulated as the OWFCR-CL problem); redundant cables are [1,9], [10,20], [19,38], [41,42] and [46,47] (in orange).

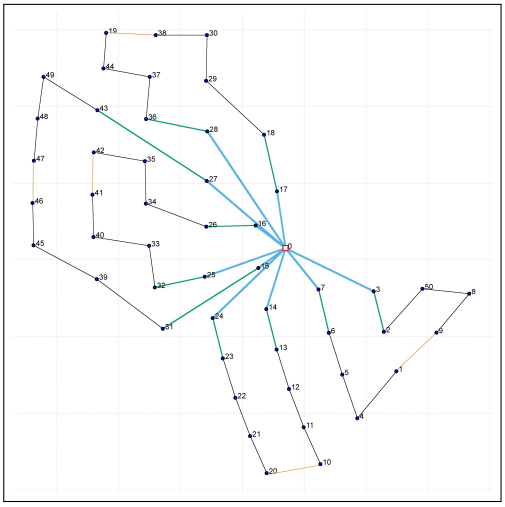


Figure 8.6 shows the solution to the OWFCR-CL problem for the considered HR3 test case. This optimal solution was found in 2931 seconds on a standard

PC. If we assume not to have any limitation on branches, the company would select this structure only to cope with cable failures. Comparing with the optimized layout for the general OWFCR problem (in Figure 8.3), this layout is 3.9 M€ more expensive, including the cost for the redundant cables (in orange in the figure). Being able to quantify the extra cost for a loop structure is a very valuable input to the business case, making it possible to the company to evaluate alternative solutions (as batteries/diesel generators attached to each turbine).

8.6 Using OTMs instead of substations (OWFCR-OTM)

The classical OWFCR problem assumes that substation(s) are fixed in advance. In very recent years, however, companies are questioning about the need for offshore substations, that are big and expensive structures involving a lot of components—while only the main transformer is required in practice. In 2015, Siemens Siemens, n.d. proposed an innovative structure, called *Offshore Transformer Module* (OTM), that is able to handle the transformer function through a smaller and cheaper hardware to be attached directly to the turbine foundations. The turbines with this OTM structure can be connected directly to shore, or to other OTM structures, through so-called *export cables*. Export cables differ from inter-array cables, in that they operate at a different voltage and have a much higher capacity (and a much larger price). Due to their different voltage, export cables cannot be connected directly to inter-array cables, but require the installation of a transformer—hence the need of the OTM. Figure 8.7 illustrates a typical cable routing involving export cables.

8.6.1 Mathematical formulation

The first modification to our MILP model (8.2)-(8.13) consists in introducing a single "dummy substation" associated with a node s located on shore, that represents the connection to the backbone electrical network. In addition, a special cable type τ is given on input, that corresponds to the export cable (with its capacity and unit cost). Also, we need to impose the following technical requirements: (a) no more than μ_1 regular cables can enter a turbine, and (b) no more than μ_2 export cables can enter each turbine. The above requirements

can easily be modeled by the following additional constraints:

$$\sum_{i \in V: i \neq h} \sum_{t \in T: t \neq \tau} x_{i,h}^{t} \leq \mu_1, \qquad h \in V_T \tag{8.25}$$

$$\sum_{i \in V: i \neq h} x_{i,h}^{\tau} \leq \mu_2, \qquad h \in V_T. \tag{8.26}$$

Finally, the fixed cost for each OTM (c_{otm}) can be added to the cost of each variable $x_{i,j}^{\tau}$, resulting in the new objective function:

$$\min \sum_{(i,j) \in A} \sum_{t \in T} c_{i,j}^{t} x_{i,j}^{t} + \sum_{(i,j) \in A} c_{otm} x_{i,j}^{\tau}. \tag{8.27}$$

Thus, the OWFCR-OTM model is the OWFCR model (8.2)–(8.14) with the addition of constraints (8.25)–(8.26), and the modified objective function (8.27). Note that our OWFCR-OTM formulation does not include the additional requirements of Sections 8.3 to 8.5, as we only want to illustrate the potential of the new technology.

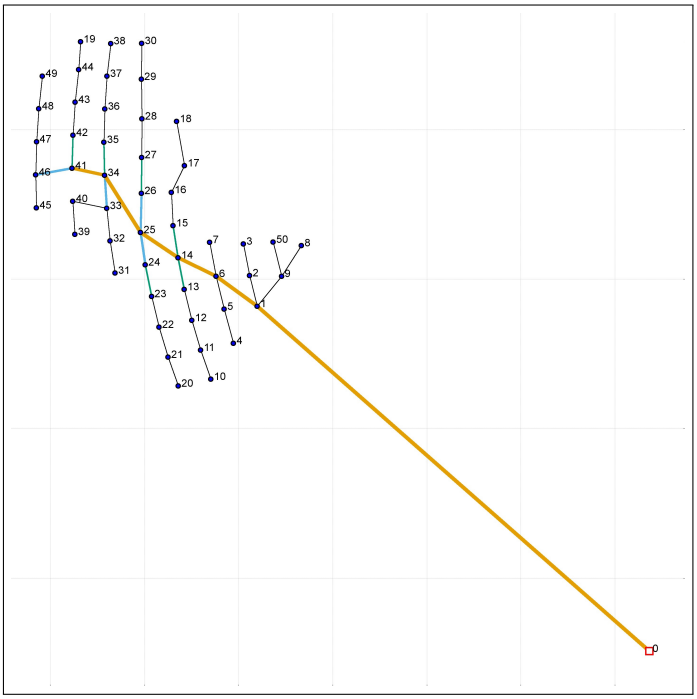
8.6.2 Example

We still use HR3 as an example of optimized solution considering OTMs. We estimated a price of 1 M€ for each OTM structure, and we set $\mu_1 = 2$ and $\mu_2 = 1$. In this case the optimization considers also an extra cable (the export cable) that can support all turbines and has a cost of 1200 €/m. Figure 8.7 show the optimized layouts using the model presented in this Section. A "dummy substation" (node 0) was located on shore.

As opposed to the previous cases, that were solved to optimality in less than one hour, this test was run until the time limit of 1 hour, ending with a gap of 12%.

We estimate the savings obtained by using OTMs instead of a substation in the following way: we fix the basic layout of Figure 8.3 and we add the cost for the export cable to shore (computed as 1200 €/m) and the substation cost of 100 M€. The layout of Figure 8.7 then allows for a saving of 101 M€.

Figure 8.7: Using OTMs allows for a saving of 101 M€.



8.7 A hybrid matheuristic/exact algorithm

The use of ad-hoc heuristics to generate an initial solution has been proven to drastically speed up the MILP solver, in particular for large size instances for which the exact solver is not able to deliver a satisfactory solution even if a large computing time is allowed. In our computational study, we used a *matheuristic* Fischetti and Lodi, 2011; Hansen et al., 2009; Fischetti and Fischetti, 2016 approach, which is a combination of mathematical programming with metaheuristics. The approach consists in designing sound heuristics on top of a black-box MILP solver, by just changing its input data in a way that favors finding a sequence of improved solutions. In our setting, the black-box MILP solver is an exact method applied to modified input.

In our hybrid scheme Fischetti and Pisinger, 2017, we first iteratively apply different refining schemes to the current best solution available. After repeating this matheuristics phase several times, we pass the best-available solution to the

MILP solver and let it run to, possibly, solve the problem to proven optimality. Our matheuristic phase works as follows: at each iteration, we temporarily fix to 1 some y variables according to a certain criterion (to be described later). Note that, by fixing to 1 some y variables, we can automatically fix to 0 all the variables corresponding to crossing arcs, thus drastically reducing the size of the problem. We then apply the MILP solver to the corresponding restricted problem, and we warm start it by providing the current solution on input. We abort the execution as soon as a better solution is found, or a short time limit of a few seconds is reached. Then all fixed variables are unfixed, and the overall approach is repeated until a certain overall time limit (or a maximum number of trials) is reached. In order to quickly find feasible solutions, we use a relaxed version of the model in the matheuristic phase. This relaxed version allows for disconnected solutions, which are however strongly penalized. More specifically, we relaxed our MILP models with some additional (continuous) slack variables, representing the current loss at each disconnected turbine. These losses are minimized in the objective function: we used a very large (big-M) cost for these loss-variables, to ensure that the optimal solution will always have no disconnected turbines. For a more detailed description, we refer the reader to Fischetti and Pisinger, 2017. The approach proved to be very effective in the first iterations of our heuristic, when even finding a feasible (connected) solution can be problematic.

Different strategies can be used to decide which y variables to fix to 1. We used the following variable-fixing criteria. Let y^* denote the best-available solution at the current iteration (y variables only).

Our first criterion simply selects with a certain probability (50%) some arcs in the current best available solution. It fixes their corresponding variables to 1 in the new solution and re-optimizes only on the remaining arcs.
Our second criterion considers *trees* of turbines defined as follows. For each arc (i, r) with $y^*_{i,r} = 1$ that enters the substation, say r, we define the node set S_i containing all the nodes that reach the substation r passing through node i. In other words, S_i contains all the predecessors of node i in the anti-arborescence corresponding to y^*. At each iteration we randomly select few trees to be optimized and we fix all the arcs not belonging to them.

Our third criterion partitions the wind farm into *sectors* of a certain angle from the substation. We then iteratively reoptimize each sector by fixing all the arcs outside the sector, i.e. all the arcs that involve two nodes not belonging to it.

The MILP models for the OWFCR problem variants were tested with our hybrid approach on the HR3 case. Most of these instances are solved to optimality

within 1 hour. Table 8.1 reports the most important figures about our HR3 runs, including gap to optimality, number of branch-and-cut nodes, number of generated user cuts (no-cross constraints), and final computing time. In the right part of the table, we report also the statistics for the same instances run using the exact method only. We imposed a time limit of 1 hour to the MILP solver (IBM ILOG CPLEX 12.6) on the same standard PC. The HR3 instance studied in our examples turned out to be an easy case, where both the exact and the hybrid approach performs similarly. According to Table 8.1, indeed, the matheuristic phase applied before the exact solver does not improve the final result for these easy instances. On the other hand, for hardest cases, it typically produces significantly better solutions within the time limit, improving the robustness (and hence the reliability) of our method. For a more detailed performance comparison between the exact and the matheuristic method, the reader is referred to Fischetti and Pisinger, 2017.

Table 8.1: Computational information about the HR3 runs

	Matheuristic+exact algorithm						Only exact algorithm					
Optimization model	LP bound [M€]	best sol [M€]	%gap	nodes	no-cross constr.	final time (sec)	LP bound [M€]	best sol [M€]	%gap	nodes	no-cross constr.	final time (sec)
OWFCR	36.69	36.69	0	4964	137	176.41	36.69	36.69	0	4736	167	43.69
OWFCR-SS	36.92	36.92	0	42796	199	598.33	36.92	36.92	0	55620	391	723.44
OWFCR-BP	36.74	36.74	0	596	19	232.52	36.74	36.74	0	979	66	15.28
OWFCR-CL	40.63	40.63	0	96649	972	2931.90	40.25	40.63	0.93	54499	1539	3600.00
OWFCR-OTM	60.80	69.31	12.27	39450	2252	3600.00	60.82	69.31	12.24	57350	2974	3600.00

8.8 What-if analysis on real-world instances

The MILP-based heuristic presented in Section 8.7 has been used on a set of real-world instances to test the economical impact of the new extensions of the model. The heuristic was programmed in C language on top of the commercial MILP solver IBM ILOG CPLEX 12.6, and was run on a standard PC with a 1-hour time limit.

8.8.1 Test instances

We tested our model on the real-world instances proposed in Fischetti and Pisinger, 2017. We considered five different real wind farms in operation in United Kingdom and Denmark, and one new wind farm under construction. These parks are named Horns Rev 1, Ormonde, Dan Tysk, Thanet, and Horns Rev 3.

Our dataset includes old and new parks, with different power ratings and different number of turbines installed, and therefore represents a good benchmark for our tests. Each park has one substation with its own maximum number of connections (denoted by C in our model).

Horns Rev 1 is one of the oldest large-scale wind parks in the world. It was built in 2002 in the North Sea, about 15 km from the Danish shore, and produces around 160 MW. Horns Rev 1 has 80 turbines Vestas 80-2 MW and $C = 10$. Our second wind farm is Ormonde, located in United Kingdom, in the Irish Sea. It has a total capacity of 150 MW (30 Senvion 5 MW turbines) and $C = 4$. Our third park is Thanet, a bigger wind park with a capacity of 300 MW (100 Vestas 90-3 MW turbines) and $C = 10$. When it was opened, in 2010, Thanet was the biggest offshore wind farm in the world. DanTysk offshore wind farm is located west of the island of Sylt and directly on the German-Danish border. With a total of 80 Siemens 3.6 MW turbines (288 MW), DanTysk can provide up to 400 000 homes with green energy. It has $C = 10$. Finally, the last layout refers to a preliminary layout for a new wind park, Horns Rev 3. Horns Rev 3 has a park capacity of 350 MW and our preliminary layout uses 50 modern big-size turbines (Vestas 8 MW) and $C = 12$. We already used this park as an illustrative example in the previous part of the paper. All the considered sites are owned by Vattenfall.

In these dataset we are also provided with different cable sets, indicated as cb01, cb02, cb03, cb04, cb05 and cb06.

Specific feasible combinations of site (i.e., wind farm) and cable set represent an instance in our testbed. Table 8.2 reports the main characteristics of instances, namely: the wind park layout (and its short name in parenthesis), the cable set name, the number of turbines in the layout, the number of cable types in the cable set, and the maximum number of connections to the substation (C).

8.8.2 What-if analysis

The possibility of quickly evaluating the economical impact of alternative design choices is considered of fundamental importance by the Vattenfall's engineers, who make several "what-if" analyses before deciding the final cable routing to be implemented. We used our real-world dataset to analyze the impact of (i) branch vs string layout; (ii) branch vs loop structure; and (iii) substation vs OTMs layouts.

Table 8.2: Main characteristics of our test instances

park	cable set	n.turb.	n.cabl.types	C
Horns Rev 1 (wf01)	cb01	80	3	10
	cb02	80	2	10
Ormonde (wf03)	cb03	30	2	4
	cb04	30	2	4
Thanet (wf04)	cb01	80	3	10
	cb04	80	2	10
	cb05	80	2	10
Dan Tysk (wf05)	cb04	100	2	10
	cb05	100	2	10
Horns Rev 3 (wf06)	cb03	50	2	12
	cb04	50	2	12

8.8.2.1 Economical value of layouts from OWFCR-SS vs OWFCR and OWFCR-BP

As already discussed, our models were developed to help the engineers of the company to evaluate the impact of different decision choices in the design of the cable networks. Previously, engineers did not have any sound optimization tool to help them, so they were often designing cable routings by strings, as it was the easier case to handle manually. Our first task was therefore to compare the string structure with possible alternatives, using our MILP models. In particular, two different situations may occur: the selected turbine model for the park can handle multiple cable connections with no extra costs, or the selected turbine model can be connected to only one entering cable by default, and extra connections can be added at an additional price. We will therefore compare optimized string layouts (from our OWFCR-SS model) with layouts from the OWFCR model, for the first case, or with layouts from the OWFCR-BP model in the latter case. In our first test we considered turbines that, by default, can be connected to at most 2 cables (one entering and one exiting), hence implementing a branch structure would imply extra costs. In this test we considered an extra cost of 25 k€ for having two cables entering a turbine ($\pi_2 = 25$ k€), and of 30 k€ for having three entering cables ($\pi_3 = 30$ k€); no extra costs was set for one entering cable, as this is the default setting ($\pi_1 = 0$). The manual operator in this case would design the routing by strings, in order to have no extra costs. We therefore compared the cost of the optimized solutions considering explicit branch penalties in the model, with the string-structure optimized layouts. Table 8.3 reports the results: the first two columns identify the test instance (park and cable set), the third column reports the cost of the optimized solution considering branching penalties explicitly in the model, while the fourth column reports the cost of the optimized solution imposing a string structure (all costs in M€). Finally, the last two columns report the

difference branch-cost minus string-cost, hence negative values correspond to savings with respect to the string structure usually implemented by planners.

Table 8.3: Optimized branch vs string structure solutions (with branch extra-costs)

park	cable set	OWFCR-BP sol. [M€]	OWFCR-SS sol. [M€]	diff [M€]	% diff
wf01	cb01	19.45	19.45	0.00	0.0
	cb02	22.62	22.62	0.00	0.0
wf03	cb03	8.08	8.13	-0.05	-0.6
	cb04	8.39	8.54	-0.14	-1.7
wf04	cb01	39.06	39.10	-0.04	-0.1
	cb04	38.77	39.64	-0.87	-2.2
	cb05	49.54	49.54	0.00	0.0
wf05	cb04	22.47	24.64	-2.18	-9.6
	cb05	26.82	27.17	-0.35	-1.3
wf06	cb03	38.70	39.60	-0.91	-2.3
	cb04	43.93	45.36	-1.43	-3.2

It can be noticed that, having an optimization tool able to explicitly consider the branch costs in the optimization, results in large savings, compared to the classical approach of using only string structures. According to our experiments, the average saving using the branch-penalty model is of about 500 k€, with extreme cases with savings of more than 2 M€ (park wf05 with cable set cb04).

Some modern turbines are constructed to handle more than one entering cable, hence no extra costs for branches are paid and the OWFCR model of Section 8.2 can be used. In this case branches become even more attractive, as shown in Table 8.4 where we report a comparison between layouts from the OWFCR model and the OWFCR-SS one when there are no costs for branches. As in the previous table, the first two columns specify the instance, the next two report the cost of the optimized solution with the OWFCR model or the OWFCR-SS model (in M€), and the last tow columns give the difference. Our results confirm that, the classical planners' approach of connecting turbines by string is way more expensive than using a branch structure. The average savings when allowing for branches is of 600 k€, with extreme cases of savings over 2 M€. Due to the large number of possible configurations, it is not trivial to manually design an optimal layout, so a sound optimization tool is needed to achieve these savings.

Table 8.4: Branch vs string structure solutions (with no branch extra costs)

park	cable set	OWFCR sol. [M€]	OWFCR-SS sol. [M€]	diff [M€]	%diff
wf01	cb01	19.44	19.45	-0.02	-0.07
	cb02	22.61	22.62	-0.01	-0.04
wf03	cb03	8.05	8.13	-0.08	-0.9
	cb04	8.36	8.54	-0.18	-2.2
wf04	cb01	38.98	39.10	-0.12	-0.3
	cb04	38.73	39.64	-0.92	-2.3
	cb05	49.35	49.54	-0.19	-0.4
wf05	cb04	22.34	24.65	-2.31	-10.3
	cb05	26.64	27.17	-0.53	-1.9
wf06	cb03	38.60	39.60	-1.00	-2.6
	cb04	43.73	45.36	-1.62	-3.7

8.8.2.2 How to handle cable failures: using generators/batteries or closed loop?

If the turbines are not equipped to survive being disconnected from the electrical network, the company can apply different strategies to limit the damage in case of failures: either to buy external batteries or generators to be connected to each turbine, or to have a closed loop structure in the cable layout. Which of these two options is the most convenient one, it is not a trivial decision. Indeed, for a planner it is not easy to manually design the cheapest closed-loop structure and to exactly quantify how much is the extra investment incurred. Table 8.5 shows the results of this test using our OWFCR and OWFCR-CL models. According to our results, adopting a closed-loop structure can be up to 5 M€ more expensive and 3 M€ more expensive on average. Having these extra costs quantified can help the engineers making a data-driven decision.

Table 8.5: OWFCR vs OWFCR-CL optimized solutions.

park	cable set	OWFCR sol. [M€]	OWFCR-CL sol. [M€]	diff [M€]	%diff
wf01	cb01	19.44	21.09	-1.65	-8.5
	cb02	22.61	24.55	-1.94	-8.6
wf03	cb03	8.05	8.68	-0.62	-7.7
	cb04	8.36	9.17	-0.82	-9.8
wf04	cb01	38.98	42.71	-3.73	-9.6
	cb04	38.73	44.23	-5.50	-14.2
	cb05	49.35	54.62	-5.27	-10.7
wf05	cb04	22.34	26.44	-4.11	-18.4
	cb05	26.64	29.77	-3.13	-11.8
wf06	cb03	38.60	43.64	-5.04	-13.1
	cb04	43.73	48.90	-5.17	-11.8

8.8.2.3 Offshore Transformer Modules or substations?

Finally, we tested the potential of the new OTM technology. To do so we considered a cost of 3 M€ for each OTM and of 1200 €/m for the export cable. The position of the onshore connection point has been estimated for each wind park looking at 4cOffshore, 2015. The estimated cost of an offshore substation is 100 M€. In these tests we assumed that the company has to take care also of the export cable costs, and that the substation position has been fixed before running our model. In our test cases we used the real substation positions for each specific park, so we can reasonably assume that its location was optimized in the design phase. Therefore, the cost of the basic-model solution can be recomputed by adding the cost of the substation and the cost for the export cable. This is why the costs of the OWFCR solutions in Table 8.6 are higher than in the previous tables. Notice that, once the position of the substation is fixed, there is no room for optimizing the capital costs related to the export cable, that are therefore just computed in a post-processing phase. The OWFCR-OTM model of Section 8.6 was used to optimize the layout using OTMs instead of substations. Table 8.6 reports the comparison of the two technologies, and shows the potential of an optimized use of OTM technology. According to the table, savings can be as large as 67%. The OTM optimized solution is, on average, 89 M€ cheaper than the classical one.

Table 8.6: Cost of OWFCR layouts (including substation and export-cable costs) vs cost of using OTMs.

park	cable set	OWFCR sol. [M€]	OWFCR-OTM sol. [M€]	diff [M€]	%diff
wf01	cb01	136.56	44.33	92.23	67.5
	cb02	139.74	51.60	88.14	63.0
wf03	cb03	149.12	54.98	94.13	63.1
	cb04	149.42	55.17	94.25	63.0
wf04	cb01	646.19	562.35	83.84	12.9
	cb04	645.94	553.94	92.00	14.2
	cb05	656.57	569.23	87.33	13.3
wf05	cb04	495.47	408.46	87.01	17.5
	cb05	499.77	420.56	79.21	15.8
wf06	cb03	172.82	81.71	91.11	52.7
	cb04	177.95	83.69	94.26	52.9

8.9 Conclusions

In the present paper we used Mixed Integer Linear Programming techniques to solve new versions of the classical offshore wind farm cable routing problem.

Thanks to our close collaboration with Vattenfall BA Wind, we have been able to investigate the most recent trends on the market and to evaluate their impact on the cable routing.

Turbines are becoming more customized, allowing them to survive being disconnected from the grid in case of failures, or even to substitute substations through the so-called Offshore Transformer Modules (OTMs). Turbine customization opens up for new possibilities in the park layout, therefore it is crucial to have an optimization tool able to quickly evaluate the economical impact of new technologies on the wind park costs. In the present paper we have introduced a flexible and reliable optimization tool, that scales well for bigger parks and more complex constraints. We have been able to handle new features in the model (i.e., closed-loop structure, non-linear branch penalties and OTMs) and to quantify their effect on real-world instances. The outcome of our tests indicates that millions of euros are involved in these analyses, so decisions based on optimized solutions can lead to substantial savings for the company and, more generally, to cheaper transition toward sustainable energy.

Acknowledgement

This work is supported by Innovation Fund Denmark. Thanks to Jesper Runge Kristoffersen, Iulian Vranceanu, Thomas Hjort, Kenneth Skaug, Urban Axelsson and Iver Slot from Vattenfall BA Wind who helped us in defining the cable routing extensions.

References

4cOffshore (2015). *website*. `http://www.4coffshore.com/offshorewind/`. Accessed: 2015-01-30.

Bauer, J. and J. Lysgaard (2015). "The offshore wind farm array cable layout problem: a planar open vehicle routing problem". In: *Journal of the Operational Research Society* 66.3, pp. 360–368.

Berzan, C., K. Veeramachaneni, J. McDermott, and U. O. Reilly (2011). "Algorithms for cable network design on large-scale wind farms". In: *Tech. Rep. Tufts University*.

Carroll, P., B. Fortz, M. Labbé, and S. McGarraghy (2013). "A branch-and-cut algorithm for the ring spur assignment problem". In: *Networks* 61.2, pp. 89–103.

Cerveira, A., A. D. Sousa, E. J. S. Pires, and J. Baptista (2016). "Optimal Cable Design of Wind Farms: The Infrastructure and Losses Cost Minimization Case". In: *IEEE Transactions on Power Systems* 31.6, pp. 4319–4329.

Dutta, S. (2012). "Data Mining and Graph Theory Focused Solutions to Smart Grid Challenges". MA thesis. University of Illinois.

Dutta, S. and T. J. Overbye (2011). "A Clustering Based Wind Farm Collector System Cable Layout Design". In: *Power and Energy Conference at Illinois (PECI)*, pp. 1–6.

Fagerfjall, P. (2010). "Optimizing Wind Farm Layout - More Bang for the Buck Using Mixed Integer Linear Programming". MA thesis. Goteborg, Sweden: Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University.

Fischetti, M. and A. Lodi (2011). "Heuristics in Mixed Integer Programming". In: *Wiley Encyclopedia of Operations Research and Management Science (James J. Cochran ed.) Vol. 8*. John Wiley and Sons, pp. 738–747.

Fischetti, M. and M. Monaci (2015). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics*, pp. 1381–1231.

Fischetti, M. and M. Fischetti (2016). "Matheuristics". In: *Handbook of Heuristics*. Ed. by R. Martí, P. Panos, and M. G. Resende. Springer International Publishing, pp. 1–33. ISBN: 978-3-319-07153-4. DOI: 10.1007/978-3-319-07153-4_14-1. URL: http://dx.doi.org/10.1007/978-3-319-07153-4_14-1.

Fischetti, M. and D. Pisinger (2017). "Optimizing wind farm cable routing considering power losses". In: *European Journal of Operational Research*. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.07.061. URL: http://www.sciencedirect.com/science/article/pii/S037722171730704X.

Fischetti, M. and D. Pisinger (2018). "Optimal wind farm cable routing: modeling branches and offshore transformer modules". In: *Network*. DOI: https://doi.org/10.1002/net.21804.

Fortz, B. and M. Labbé (2004). "Two-Connected Networks with Rings of Bounded Cardinality". In: *Computational Optimization and Applications* 27.2, pp. 123–148.

González, J. S., M. B. Payán, J. M. R. Santos, and F. González-Longatt (2014). "A review and recent developments in the optimal wind-turbine micro-siting problem". In: *Renewable and Sustainable Energy Reviews*, pp. 133–144.

González-Longatt, F. M. and P. Wall (2012). "Optimal Electric Network Design for a Large Offshore Wind Farm Based on a Modified Genetic Algorithm Approach". In: *IEEE Systems Journal* 6.1, pp. 164–172.

Hansen, P., V. Maniezzo, and S. Voß (2009). "Special issue on mathematical contributions to metaheuristics editorial". In: *Journal of Heuristics* 15.3, pp. 197–199.

Hertz, A., O. Marcotte, A. Mdimagh, M. Carreau, and F. Welt (2012). "Optimizing the Design of a Wind Farm Collection Network". In: *INFOR* 50.2, pp. 95–104.

Kis-orca (2015). *website*. `http://www.kis-orca.eu/downloads/`. Accessed: 2015-01-30.

Labbé, M., G. Laporte, I. R. Martín, and J. J. S. González (2004). "The Ring Star Problem: Polyhedral analysis and exact algorithm". In: *Networks* 43.3, pp. 177–189. ISSN: 1097-0037.

Li, D., C. He, and Y. Fu (2008). "Optimization of Internal Electric Connection System of Large Offshore Wind Farm with Hybrid Genetic and Immune Algorithm". In: *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT2008)*, pp. 2476–2481.

Pillai, A., J. Chick, L. Johanning, and M. K. V. D. Laleu (2015). "Offshore wind farm electrical cable layout optimization". In: *Engineering Optimization* 47.12, pp. 1689–1708.

Qi, W., Y. Liang, and Z.-J. M. Shen (2015). "Joint Planning of Energy Storage and Transmission for Wind Energy Generation". In: *Operations Research* 63.6, pp. 1280–1293.

Sannino, A., H. Breder, and E. Nielse (2006). "Reliability of collection grids for large offshore wind parks". In: *Proceedings of 9th International Conference on Probabilistic Methods Applied to Power Systems*.

Siemens (n.d.). *New AC Grid Access Solution from Siemens: Lighter, faster, cheaper*. `http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2015/energymanagement/pr2015030151emen.htm`.

Zhao, M., Z. Chen, and F. Blaabjerg (2009). "Optimisation of Electrical System for Offshore Wind Farms via Genetic Algorithm". In: *IET Renewable Power Generation 3* 3.2, pp. 205–216.

Part IV

Usage in Vattenfall

Chapter 9

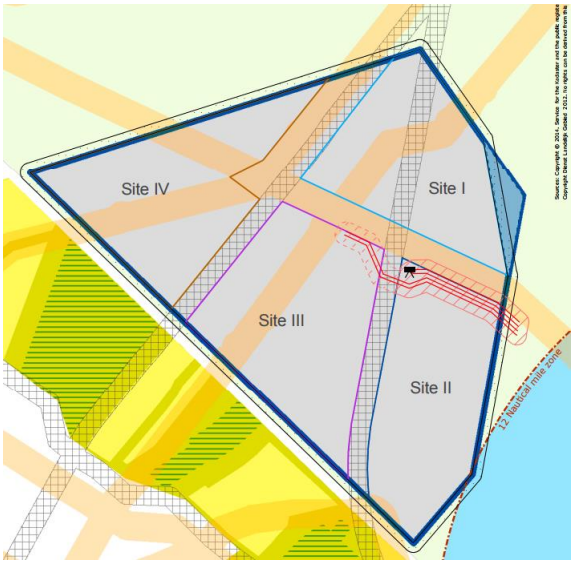
Testing the wind farm layout optimizer on a real project

In this chapter we will discuss our personal experience in using the wind farm layout optimization tool on a real-world case. This project was used, inside Vattenfall, as a test for the optimization framework we developed (see Part II of the present thesis). We decided to report this experience in the thesis to give the reader a better overview of how the wind farm design process is handled in practice, and what kind of what-if analyses are requested. We tested our tool against the commercial software in use in the company (windPRO). This software has both an optimization module and a simulation one. The optimization module uses heuristic techniques to locate turbines in the site to maximize power production – more specifically, Annual Energy Production (AEP). The WindPRO optimizer (as well as no optimizers known by the company) is not able to explicitly consider costs of foundations in the optimization. For this reason, we will first test our wind farm layout optimization framework considering only power production. We will thus compare our results with the commercial software ones, under the same assumptions. As a second step, we will introduce cost of foundations in the optimization and test the impact of this in the final layout. The windPRO simulator will be used to evaluate our layout and its own, in order to have a fair comparison.

9.1 Our test site

The wind farm area we considered for testing our optimization framework is located at the southern border of The Netherlands and it is called Borssele. The wind farm area of approximately $344km^2$ is sub-divided into four wind farm sites. The different sites were put on tender separately. [1] We will focus on site I, which is the the north-eastern one, as shown in Figure 9.1. For each area, the total wind farm capacity is 350 MW. This limit is actually of 380 MW installed power, but the produced power should never exceed 350 MW, therefore turbines may be curtailed or stopped in case this limit is reached.

Figure 9.1: The whole area to construct the wind park is divided in four sub-areas (sites)

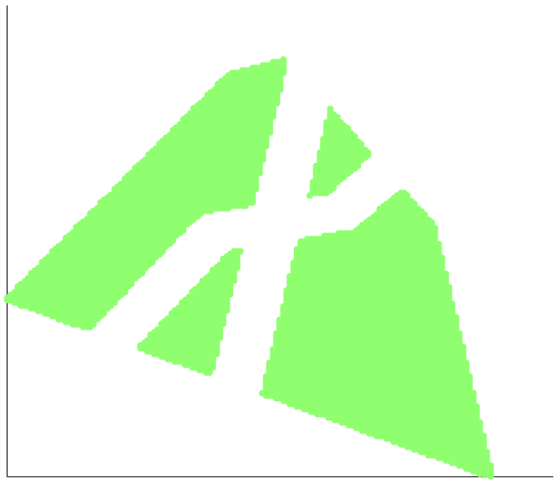


We obtained 7890 possible turbine positions in site I by over-imposing a grid of 75 x 75 m. At the time, we decided that this was a good trade-off between model size and grid size, considering that modern offshore turbines can have a rotor diameter of over 150 meters. Nowadays, we use a more refined grid in our site optimization, since we experienced that our tool scales well with the number of possible turbine positions in input. Our possible positions on input excluded different no-go areas. In real cases, no-go areas can be of different

[1] Sites I and II (for a combined 700-760 MW capacity) were awarded to DONG Energy in July 2016, while sites III and IV (with a further 680-740 MW) were awarded to Shell in September 2016.

nature, such as archaeological findings, existing infrastructure, natural reserves, and so on. In this case the excluded areas refer to two cables already existing in the area: turbines cannot be built within a certain buffer from these cables. The resulting available area is the site in Figure 9.2.

Figure 9.2: Available area to locate turbines



Wind time series for the site have been used in order to give to the tool an understanding of the wind in the site. This data has been preprocessed in order to define different wind scenarios. A wind scenario is defined as a couple (direction, wind speed) and is associated to a given frequency (we refer to it as probability in our model, see Chapter 6.

In our site we considered wind measurements sampled every 10 minutes for 20 years. A preprocessing algorithm partitions them considering the direction of the wind measured and its intensity. The probability is obtained by counting how many measurements fall in each bin. In this experience we noticed the importance of having enough bins, especially in wind direction (see A.3). We binned the wind samples in 0.1 degree, 1 m/s bins. All in all, we considered about 70000 wind scenarios for this case.

9.2 Tests on wake effect

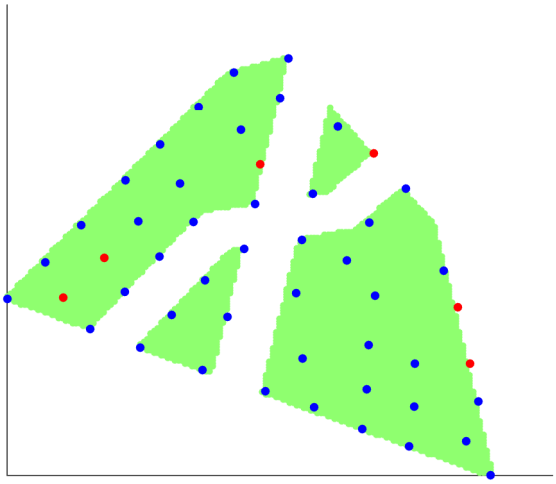
The first tests were run to compare the optimized layout with the one provided by Vattenfall's experts (computed using the commercial software windPRO). Since the commercial software optimizes only on park production, we will here do the same, not considering the cost of installing the turbines (see next section for optimization including installation costs).

The objective of the optimization tool is to maximize the power production considering the wake effect (so maximize power minus interference) while satisfying the constraints. When evaluating the layouts through simulations WindPRO evaluates also *wake efficiency*. This is defined as the wake reduced production compared to the unobstructed production. Wake efficiency is not considered explicitly in our model, since the tool aims for the maximum energy production even if this could imply an increase in the wake effect. An easy example of this is considering two layouts with a different number of turbines: as to AEP, according to the tool, a solution with more turbines will always win (since it produces more power) even if wake losses would probably increase (so wake efficiency decreases) in this case. Nevertheless, in the practical usage of the optimizer, the number of turbines is typically fixed on input. In this setting, the objective of the optimization tool is the most interesting one (and wake efficiency is a natural consequence). Therefore, in the following comparison with windPRO, we will record both AEP (that is the equivalent of our objective function but expressed in MWh/y) and wake efficiency. It will be clear that, having a fixed number of turbines, our optimizer implicitly optimizes wake efficiency as well.

For our test case we are willing to locate 50 7 MW (rotor diameter of around 150 m) in the possible positions of Site 1. In this first test, we artificially set water depth equal to zero in all the site, in order to have a layout optimized only on production. We will compare the output with the company layout (also designed only on wake effect considerations) using an external AEP calculator (namely WindPRO). First, we will look at the WindPRO solution (in Figure 9.3). It locates 50 turbines in the available area: most of them at 7 rotor diameter of distance, 6 of them at a smaller distance (6 rotor diameters). We will refer to this layout as LBOR_WP. Notice that the minimum distance between turbines is normally set to 5 or more rotor diameter as a rule of thumb to reduce wake effect and turbulence between turbines. In the past, indeed, most of the layouts were designed manually and the idea that the further the turbines the less the interference, was simply translated in a minimum distance between turbines. Old parks were designed manually by simply putting turbines on a grid at a distance of, for example, 7 rotor diameters (depending from site to

site). This rule of thumb is still very much used by practitioners nowadays. It can be noticed from the LBOR_WPexample, that locating turbines at a given minimum distance may not be a trivial task, and in this case some turbines are located closer than 7 rotor diameters (red dots in Figure 9.3).

Figure 9.3: Layout LBOR_WP from WindPRO. Blue turbines respect the 7 rotor diameter distance while red turbines are located closer



We first ran our optimization tool with a maximum number of turbines equal to 50 and a minimum distance of 7 rotor diameters. Note that for our model these are hard constraints, so a solution with turbines closer than 7 rotor diameter would not be considered feasible. We run the optimizer for 10 hours, obtaining the solution in Figure 9.4, that we will call LBOR_MF1.

Afterwards we ran the optimization with a maximum number of turbines equal to 50 and a minimum distance of 5 rotor diameters. This means that the tool is now allowed to locate turbines closer one to each other (up to 5 rotor diameters), if this is convenient even when considering wake effects. In this case the layout LBOR_WPwould be a feasible layout for our model. We ran the optimizer for 10 hours, obtaining the solution in Figure 9.5 , that we will call LBOR_MF2.

Table 9.2 compares the 3 layouts. Their value is computed using WindPRO simulation tool.

Figure 9.4: LBOR_MF150 7MW turbines at a minimum distance of 7 rotor diameters. Colors in the background represent interference over all the possible wind scenarios on input, considering their frequency.

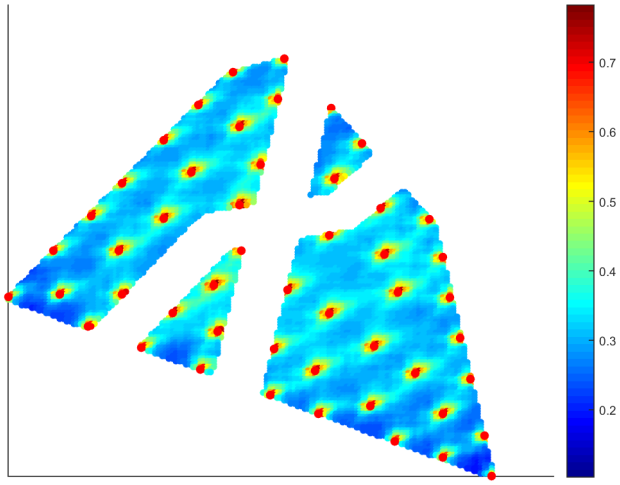


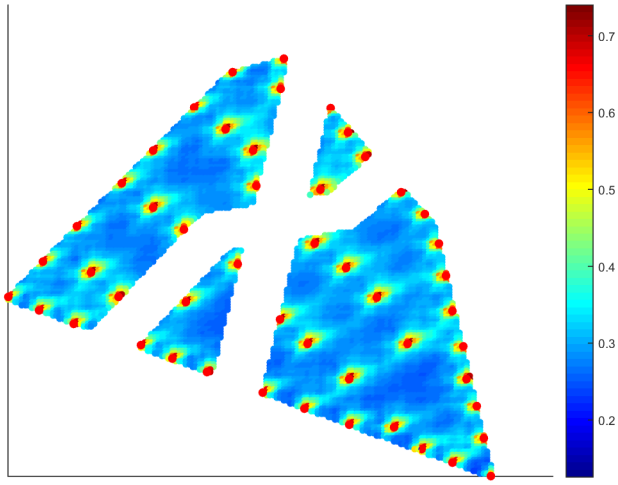
Table 9.1: Comparison of the different layouts for site I (optimized on wake effect only).

	LBOR_WP	LBOR_MF1	LBOR_MF2
AEP [MWh/y]	1 543 262	1 547 454	1 552 035
Wake efficiency [%]	93.59	93.75	94.02
Ranking based on AEP	3	2	1

Table 9.2 shows that both the layouts found by the optimization tool outperform the layout designed by the commercial software previously used in the company. Figure 9.6 shows a direct comparison between LBOR_MF2and LBOR_WP. The optimized layout (LBOR_MF2) beats the Vattenfall's layout (LBOR_WP) by 0.57% AEP (about 8800MWh/y more in production), which translates in about 600 k€/y increased revenue considering an average electrical price of 0.69 €/kWh.

It is interesting to notice that the best layout out of the three is the one imposing the less minimum distance, contrary to the rule of thumb we discussed. This is because the model, in this case, is free to evaluate more solutions and find the best one in a larger solution space. Differently from a human plan-

Figure 9.5: Layout LBOR_MF2: 50 7MW turbines at a minimum distance of 5 rotor diameters. Colors in the background represent interference over all the possible wind scenarios on input, considering their frequency.

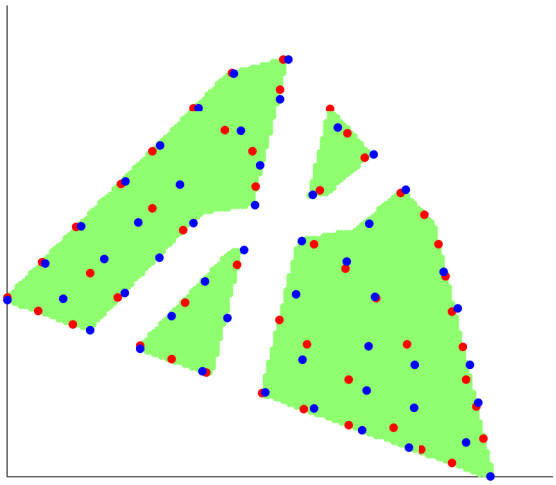


ners, indeed, the software can compute interference explicitly while optimizing. Note that, we still consider a minimum distance greater than one rotor diameter (usually of 4 or 5 rotor diameter) to accommodate for turbine loads, that are not yet considered in the optimization. Very interesting is also to observe the shape of the best layout, that tends to maximize the use of the borders of the area, where a turbine causes interference to fewer other turbines. This is a new shape for wind farms (compared with the classical grid one), able to reduce the overall power losses due to the wake effect.

9.2.1 Example of what-if analysis: exclusion of some areas

Having such good results on a simple test case, motivated our team to run some what-if analyses with our model. For example, we considered the effect of excluding some erosion zones and extreme water depth (more than 32.5 m). To do so, we simply removed these positions from the possible input positions, reducin the total number of possible locations from about 7800 points to about

Figure 9.6: LBOR_WP(blue) vs LBOR_MF2(red)



2750 points. Figure 9.7 shows the available area.

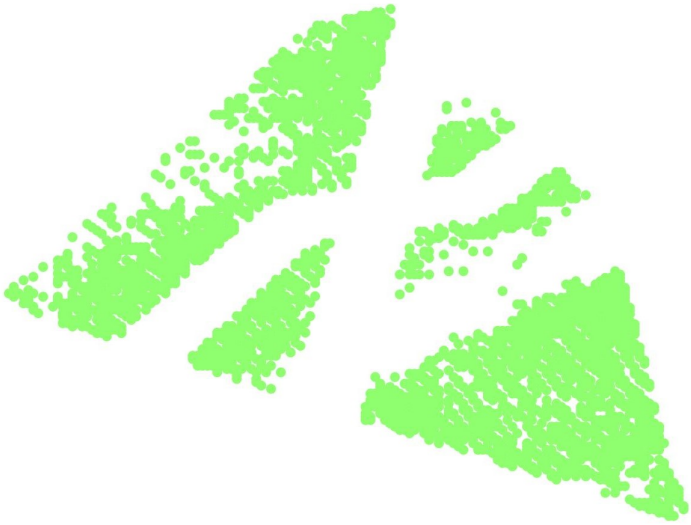
We re-ran the optimization tool as before (imposing to build the same 50 turbines with a minimum distance of 5 rotor diameters). Figure 9.8 shows the new optimized layout, that we called LBOR_MF3.

Reducing the possible areas where turbines can be built, we reduced the degree of freedom of the optimization tool, therefore we expect to find a worse solution compared with the unconstrained problem. Nevertheless, also the new layout LBOR_MF3, outperforms LBOR_WP(of about 0.32% in AEP). Table 9.2 summarizes the results.

Table 9.2: Comparison of the different layouts. The reported values have been computed by WindPro.

	LBOR_WP	LBOR_MF1	LBOR_MF2	LBOR_MF3
notes	Vattenfall	min dist 7 rd	min dist 5 rd	5 rd and erosion zones
AEP [MWh/y]	1 543 262	1 547 454	1 552 035	1 548 263
Wake efficiency [%]	93.59	93.75	94.02	93.79
Ranking based on AEP	4	3	1	2
improvement wrt LBOR_WP[MWh/y]	0	4 191	8 773	5 001
improvement wrt LBOR_WP[%]	0	0.27	0.57	0.32

Figure 9.7: Available area to locate turbines excluding erosion zones and water depths larger than 32.5 meters



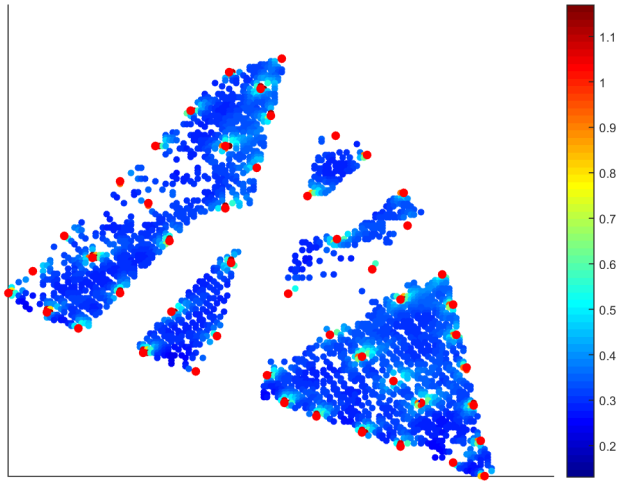
The results prove the effectiveness and the potential of using a sound optimization tool to define wind farm layouts.

9.3 Optimizing including cost of foundations

The Borssele project was characterized by very different water depths in the area of interest. For this reason, it has been considered of major interest to include costs of foundations in the optimization. This optimization feature, to the best of our knowledge, is not available in any commercial software.

We used the cost function as defined by Vattenfall experts and external consultants for this specific project. In this case, only monopile (MP) foundations are considered but, depending on the water depth, they will need a different amount of steel (and therefore have a different price). We also considered a fixed cost for the transition piece (TP) to be added to this value.

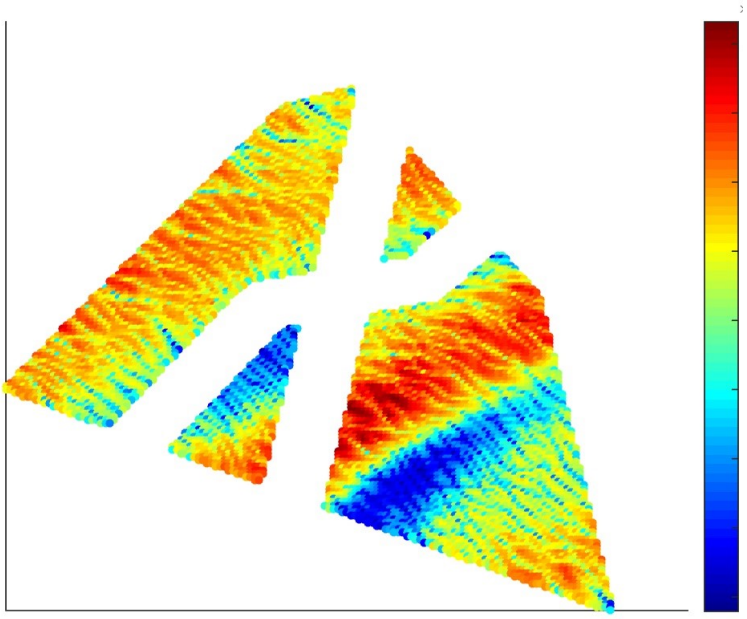
Figure 9.8: layout LBOR_MF3: optimized layout excluding erosion zones and water depths larger than 32.5 meters



Different positions in the site, depending on their water depth, will therefore correspond to different foundation prices (shown in different colours in Figure 9.9).

The optimization tool has been adapted in order to receive a site-specific cost map on input, in order to be as flexible as possible to be used in different projects. The cost of foundations is then inserted in the objective function so that the tool will optimize turbine positions considering both the revenue due to power production over 20 years of operation, and the initial cost of installation according to the input map. In order to properly compare production (MWh/y) with foundation costs (Euro), a discounting factor has been provided by Vattenfall's experts. As this value is very project-specific, it has been set as an input parameter for the tool. In our case we used the real site-specific discount factor as computed by Vattenfall's experts, that is the net present value for one MWh of production over 20 years (considering WACC and subsidies).

Figure 9.9: Cost map for Borssele site 1: different colors represent different foundation costs, from blue (the cheapest positions) to red (the most expensive ones). The exact values are hidden due to privacy issues



9.3.1 The effect of considering cost of foundations

We first wanted to test how our tool was considering the foundation costs (since we did not have any direct comparison with a commercial tool). In order to have a feeling of the effect of considering cost of foundations directly in the optimization, we asked the optimizer to locate a fixed number of turbines considering or not the wake effect in the optimization. All in all, we considered

- Site 1
- Cost of foundations from input map
- Wind climate in the area (to compute production and wake losses)
- 7MW turbines

- A fixed minimum distance of 5 rotor diameters

We asked the tool to optimize the layout locating 5, 10, 25 or 50 turbines, first looking only at foundation costs (first column in Figure 9.10) , and then balancing between foundation costs and wake effect (second column in Figure 9.10). When running the optimization disregarding wake effect, we expect that the tool will locate as many turbines as possible (within the constraints) in the less expensive areas. As shown in the first column of Figure 9.10, this is exactly what the tool is doing, proving the correctness of the optimization. On the other hand, when we introduce wake effect in the optimization, we start the challenging part that could not be done (and was never done in such a way, to our knowledge) manually. The optimization is indeed now balancing the increased income due to the selection of less-waked positions, and the possibly increased price of foundations. To balance the value of these two factors, we used here a site-specific discounting factor. As it can be observed in the second column of the figure, the effect is that the turbines are not all concentrated in the less expensive areas as before, but are now more spread in order to reduce the wake effect.

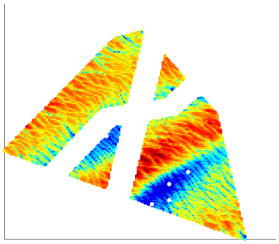
The second test is then to compare the optimized solution balancing AEP and foundation costs, with the two extreme options (considering only foundation costs, and considering only AEP). Figure 9.11 compares the three results in the case of 50 turbines.

It can be noticed that the driver is still the production since the layout considering both wake effect and foundation costs (in the Figure 9.11b) is closer to the one considering only AEP (in the Figure 9.11c) than to the one considering only foundation costs (in the Figure 9.11a). Still, the optimal layout changes when considering costs of foundations (see also Figure 9.12 for a direct comparison of layout 9.11b and 9.11c).

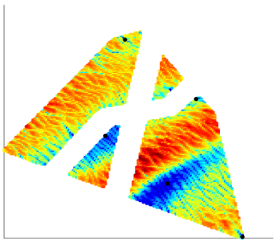
The solution optimized only on wake effect produces 3616 MWh/y more but is 9 M€ more expensive in foundation costs. All in all, considering the net present value for a MWh production over 20 years (i.e., the discounting factor) given for this specific project, the layout optimized considering foundation cost is 7 M€ (NPV) more valuable.

Note that this comparison is done between two optimized solutions (both computed with the same tool, just with different settings). The savings would be much higher if we compare with a manually designed layout. We will next compare our optimized layout considering both wake effect and foundation costs, with this company layout LBOR_WP(Figure 9.14).

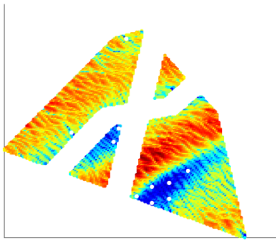
Figure 9.10: Comparison between layouts optimized by considering only foundation costs (left) or by considering both wake effects and foundtion costs (right) for 5 – 10 – 25 or 50 turbines to build



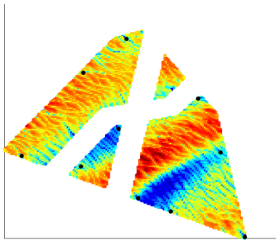
(a) No interference, 5 turbines



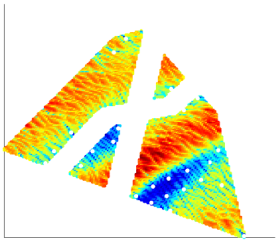
(b) With interference, 5 turbines



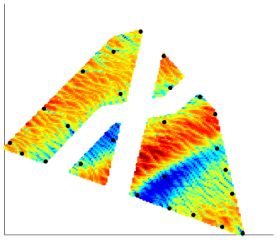
(c) No interference, 10 turbines



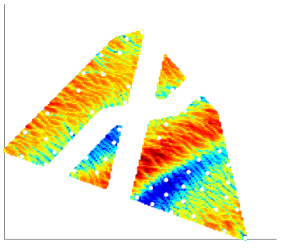
(d) With interference, 10 turbines



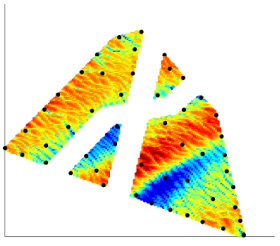
(e) No interference, 25 turbines



(f) With interference, 25 turbines

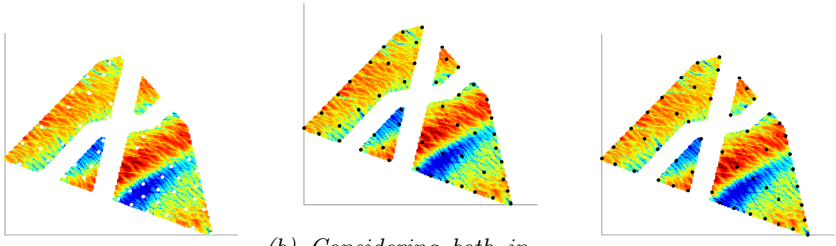


(g) No interference, 50 turbines



(h) With interference, 50 turbines

Figure 9.11: The impact of considering/not considering costs of foundations or wake effect when designing the layout

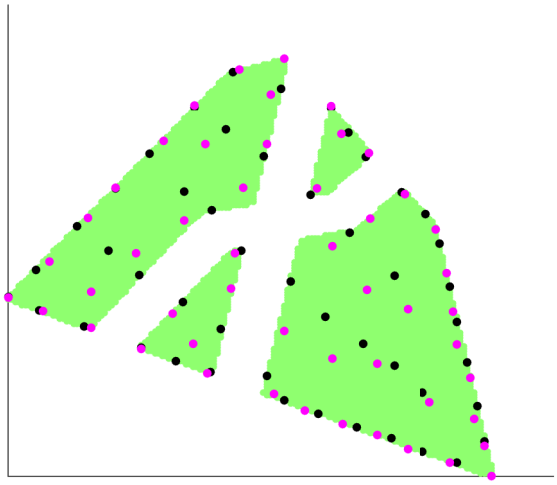


(a) Considering only cost of foundations, 50 turbines

(b) Considering both interference and cost of foundations, 50 turbines

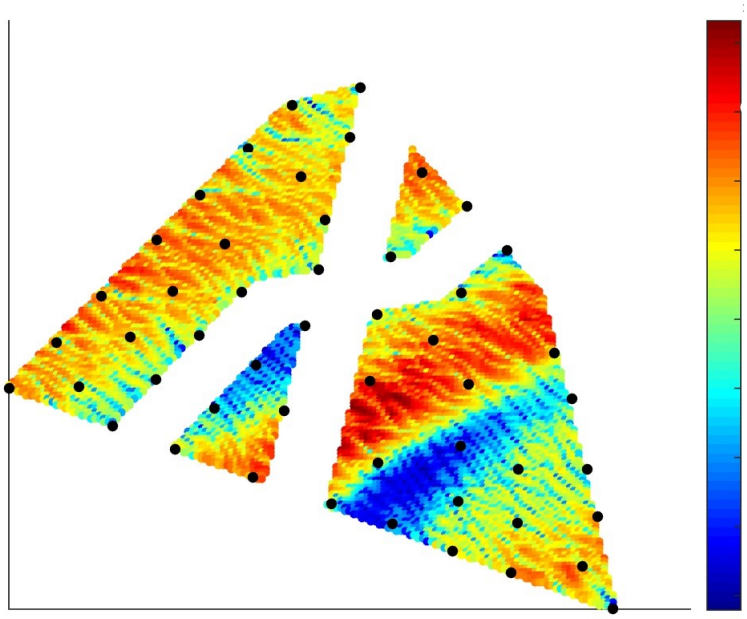
(c) Considering only interference, 50 turbines

Figure 9.12: Black dots correspond to the optimized layout considering both wake effect and costs of foundations. The pink one corresponds to the one optimized only on wake effect (and no foundation costs)



The final solution (considering both foundation costs and wake effects) has been evaluated by the project-specific team. They verified that our layout allows for an extra 0.28% production compared with LBOR_WP, while also decreasing the

Figure 9.13: Vattenfall's layout LBOR_WP. Colors in the background represent cost of foundations



price of foundations of more than 10 M€. All in all, we estimated an increased income of 12.4 M €in 20 years (NPV). Not all the area in the site was actually available for building turbines and some areas needed to be excluded. We therefore repeated our optimization on a restricted area (as described in next Section).

9.3.2 Considering additional restrictions

Some areas in the site are erosion or sedimentation zones. In these areas it is not advisable to build turbines. We therefore excluded these areas in a pre-processing phase. The resulting cost map for the reduced are is shown in Figure 9.15. The same cost function as before has been used to compute this map.

We repeated our optimization considering foundations costs and wake effect

Figure 9.14: Optimized layout considering wake effect and costs of foundations (black) versus LBOR_WP(pink)

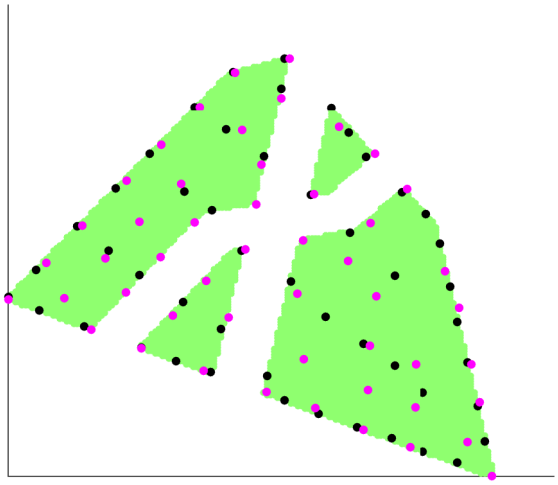
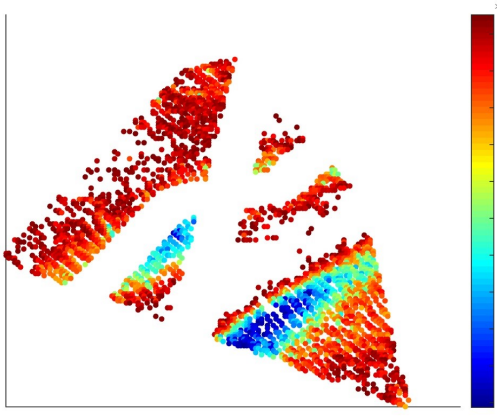
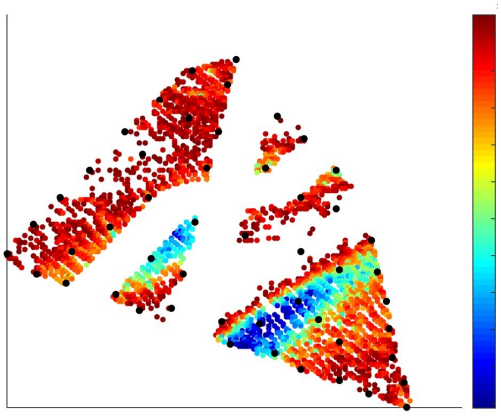


Figure 9.15: Restricted area to be considered in the optimization (colors refer to the cost map on input)



on the new input map, obtaining the layout in Figure 9.16. We will call this layout LBOR_MF_CM.

Figure 9.16: Optimized layout considering wake effect and costs of foundations (black) – LBOR_MF_CM

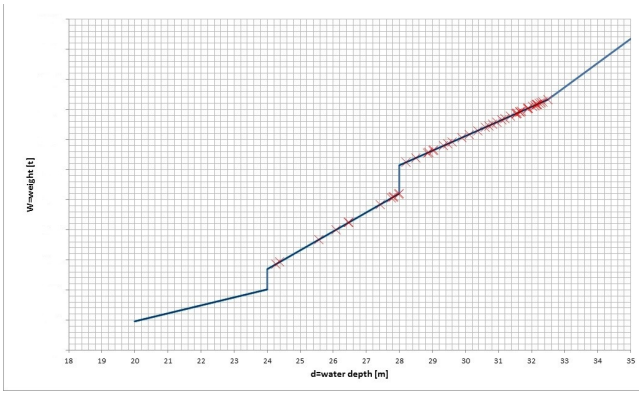


The AEP calculation by the team shows still an increase of + 0.24% AEP, resulting in a 4.5 M €increase in NPV, compared with LBOR_WP. In Table 9.3 we record some more informations about the water depth of the selected positions in the optimized layout vs the company one. In Figure 9.17 we show the amount of steel [t] needed for the foundations of the company layout and the one used for our layout. It can be notice that, by optimizing considering foundation costs, the turbines are better distributed at different water depth, decreasing the total amount of steel needed to construct their foundations (and therefore their cost).

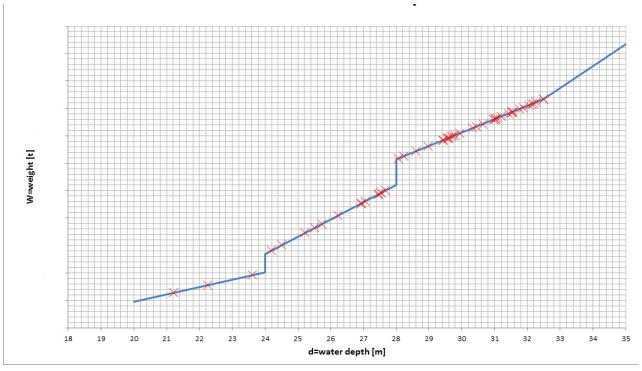
Table 9.3: Water depth comparison between the company layout (LBOR_WP) and the optimized considering foundation costs (LBOR_MF_CM)

	water depth LBOR_WP	water depth LBOR_MF_CM
max	32.48	32.49
min	24.27	21.21
avg	29.9	28.95

Figure 9.17: Amount of steel used to construct LBOR_WPor our optimized layout (LBOR_MF_CM). The blue lines represent the picewise linear function used to evaluate the foundation weight. The red crosses represent the weight for each of the selected positions in the layout. The y-axis values are not shown for privacy issues.



(a) LBOR_WP



(b) LBOR_MF_CM

Acknowledgments

We would like to thank Stathis Koutoulakos, Gijs Nijsten, Liam Murray and Victoria Gómez Ruiz who collaborated with us in this test phase.

References

EMD (n.d.). *WindPRO*. http://www.emd.dk/windpro/frontpage.

Fischetti, M. and M. Monaci (2016). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: 10.1007/s10732-015-9283-4. URL: https://doi.org/10.1007/s10732-015-9283-4.

Jensen, N. (1983). *A note on wind generator interaction*. Tech. rep. Technical Report Riso-M-2411(EN), Riso National Laboratory, Roskilde, Denmark.

Mortensen, N. G. (2010). *Wind farm AEP and wake loss calculations*. Tech. rep. Risø DTU, Denmark.

Siemens (n.d.). *New AC Grid Access Solution from Siemens: Lighter, faster, cheaper*. http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2015/energymanagement/pr2015030151emen.htm.

Chapter 10

Usage of the optimizers in real projects

10.1 Introduction

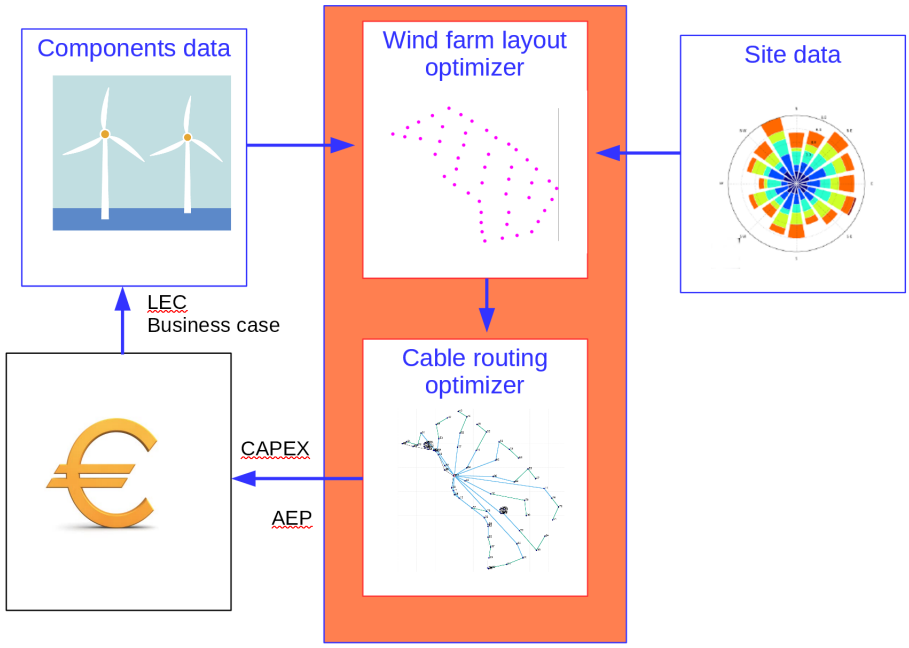
After a testing phase (see Chapter 9) the wind farm layout and the cable routing optimizers are now used in practical projects inside Vattenfall. The two tools are used in sequence (first defining the turbine layout and then the cable routing) and their output is post-processed to be used in the business case.

Figure 10.1 gives an overview of how optimization is used in the business cases:

- site related data (including wind statistics) and turbine characteristics are given on input to the wind farm layout optimizer;

- the resulting layout is used to design the cable routing, through our second optimizer;

Figure 10.1: Schematic overview of the usage of our optimization tools in Vattenfall



- the resulting layout and cable routes are provided on input to internal models to better evaluate their value;
- this evaluation is used to understand how to improve the input data and constraints;
- the process is iterated for different what-if analyses, in order to find the overall best park design.

It is very important for Vattenfall, indeed, to be able to evaluate different potential scenarios in an efficient way. In this chapter, we will refer to a *scenario* as the full collection of assumptions needed to evaluate a business case. This includes all the input values to our optimizers (what turbine model should be used, the site restrictions, the wind data, what types of cable, what routing configurations) but also the input value for the economical model that post-process our results. This post-processing model, indeed, includes also the costs

for operating the park, depending on the maintenance strategy or the estimated availability losses for the specific site. Annual Energy Production (AEP), immediate costs (CAPEX) and operational costs (OPEX) are used together to evaluate each possible scenario, through the so-called Levelized Energy Cost (LEC). LEC is the net present value of the unit-cost of electricity over the lifetime of the park, and is computed as the ratio between all the costs and the energy production (i.e., the revenue) over the park lifetime. Changing the input to the three different models (our two optimizers and the business evaluator) one can efficiently test different scenarios and evaluate the impact of different design choices.

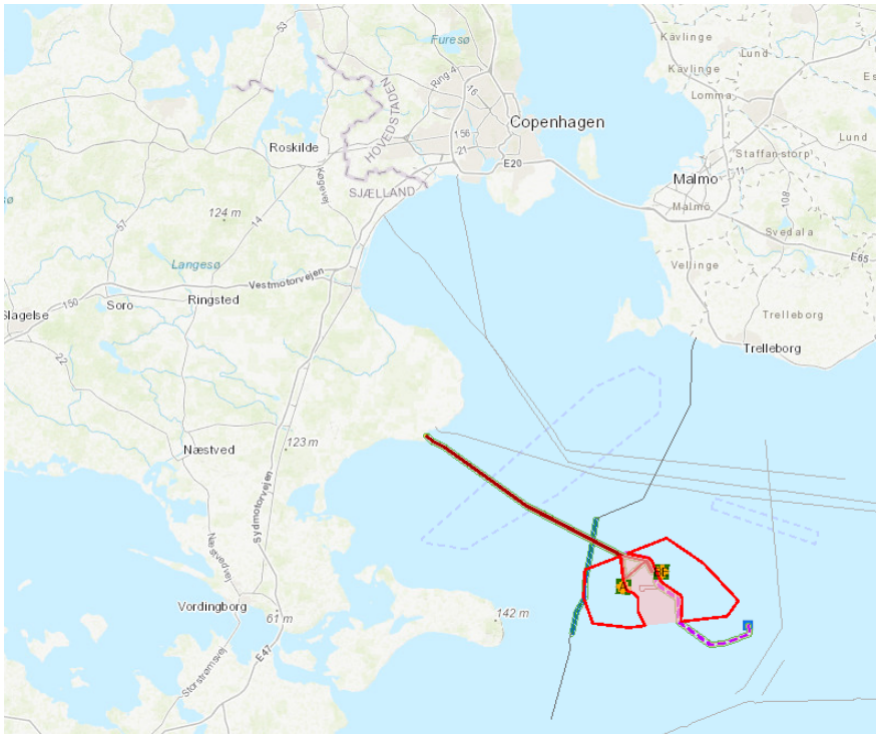
10.2 Danish Kriegers Flak

Our optimization models have been used, among others, for the design of the Danish Kriegers Flak (DKF) wind park. DKF is a 600 MW offshore wind farm in the Baltic Sea, located in a very favourable area. In addition to good wind conditions and a water depth between 16 m and 25 m, DKF is located close to the German offshore wind farm EnBW Baltic 2. Kriegers Flak will take advantage of this and will be connected both to the Danish grid as well as to the German one, through EnBW Baltic 2 and EnBW Baltic 1. The wind farm consists of two partitions, a western one one of 200 MW covering 69 km^2, and an eastern of 400 MW covering 110 km^2. Figure 10.2 shows the DKF area (boundaries in red) and its connections to the grids (in dark red the connection to the Danish grid, and in purple the connection to the nearby German park). In November 2016, Vattenfall won the tender to build Danish Kriegers Flak. The winning bid of 49.9 € per MWh was among the lowest costs in the world for offshore wind power.

In order to optimize DKF we had to take a closer look at the available area. Due to specific tender rules, not all the area was actually available but only part of it (in green in Figure 10.4). Inside the area there are also some small obstacles (green dots in Figure 10.3), that refer mainly to archaeological restrictions given by Danish Energy Association (DEA). Those points are removed from the available positions for the layout optimizer.

The two sites have been optimized at the same time, considering also the interference from the existing turbines of EnBW Baltic 2. In this case, we had 16 000 possible positions for locating turbines and about 43 000 wind scenarios (obtained by grouping real-world measurements in bins of 0.1 degrees and 1m/s). One scenario was to locate 72 8.4 MW turbines (24 in the west and 48

Figure 10.2: Danish Krigers Flak wind farm location



in the east site) with a minimum distance of 5 rotor diameters. In this case, we ran the optimizer for 24 hours on a standard PC. Every time a new incumbent was found, the new solution was temporarily stored (and in this case plotted in Figure 10.5). It can be seen how the optimizer using the proximity search, quickly improves at the beginning and it saturates on a (probably) optimal solution in about 10-15 hours. Figure 10.5 shows the evolution of the solution over time: on the top of each plot the current layout is shown (yellow dots), while on the bottom the evolution of the solution value is plotted. The aim of the optimizer is to maximize park production (AEP in the plot), so the higher the AEP the better. Given the evolution of the solution value over time, the optimizer is normally run for 8-10 hours. Note that, in the layout optimizer, we cannot rely on CPLEX gap to prove the optimality of the solution, since we changed our objective function according to the proximity search recipe. Nevertheless, the fact that the MILP solver fails at finding an improved solution for a long time is a good indicator that the solution at hand is a high-quality one.

Figure 10.3: Danish Krigers Flak wind farm area (within the red boundaries) and the existing nearby park (EnBW Baltic 2, blue dots)

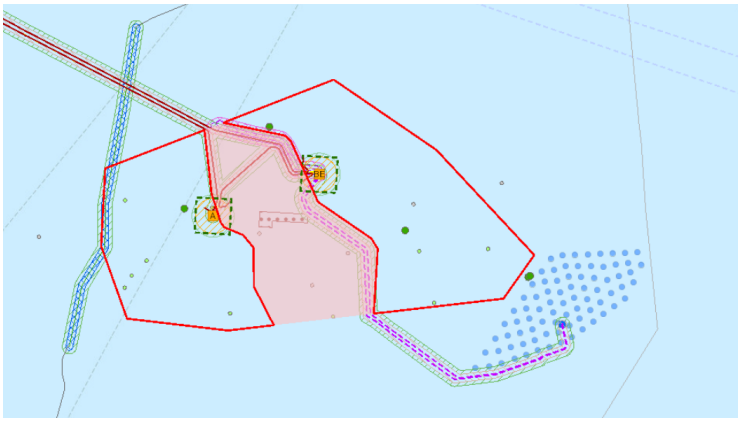
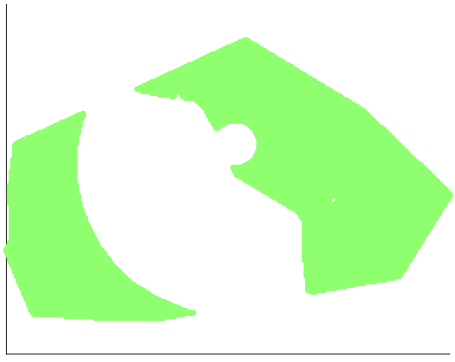


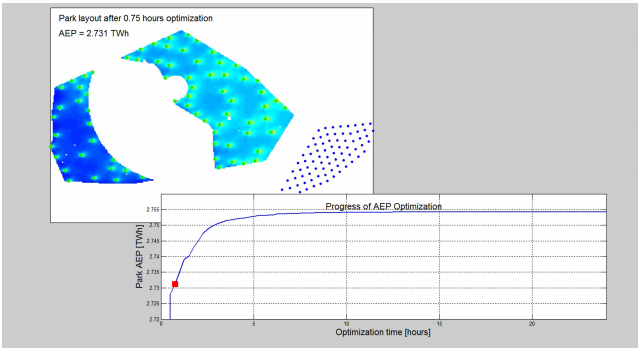
Figure 10.4: Available area to construct turbines (in green)



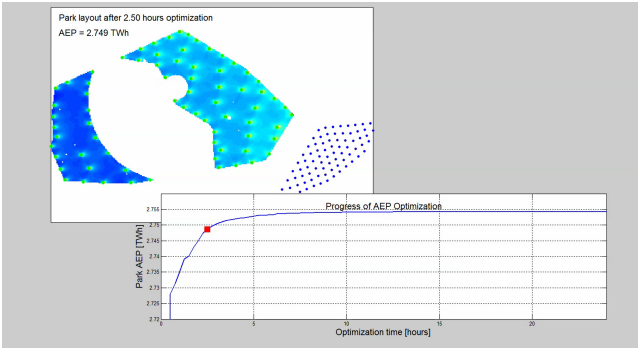
The final layout is shown in Figure 10.6: red crosses indicate the optimized position for Vattenfall's turbines, while the blue crosses show the position of the EnBW Baltic 2 turbines. The EnBW Baltic 2 turbines are 3.6 MW turbines, and their interference is considered in the turbine optimization.

Fixed the layout, we designed the cable routing using our second optimization. We considered two kinds of cable (one supports 2 turbines, the other up to 4), including power losses in our optimization. We were asked to connect turbines in strings (no cable branches allowed). Since each of the two sites has its own

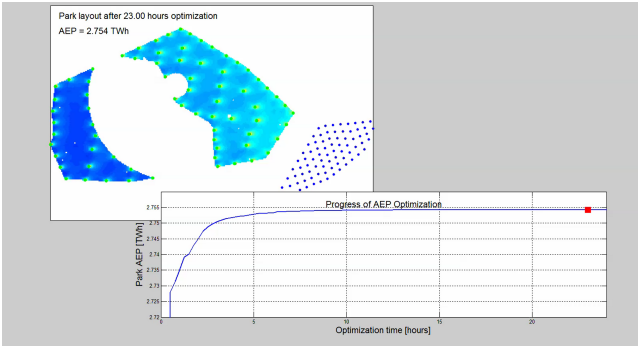
Figure 10.5: Evolution of the solution quality over time. Each plot records the status of the optimization at a given time: on the top the best-layout-so-far is plotted (yellow dots are turbines, background colors refers to interference values); on the bottom the evolution of the solution quality (the higher the better) is shown.



(a) DKF optimize layout after approx. 1 hour

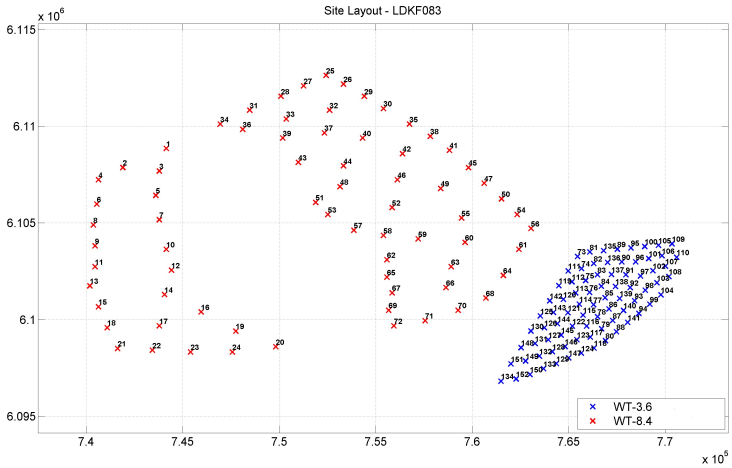


(b) DKF optimize layout after approx. 3 hours



(c) DKF optimize layout after 23 hours

Figure 10.6: Optimized layout for Danish Krigers Flak (in red), considering the interference from an existing park nearby (blue)



substation (given on input), we optimized the two sites independently. In this case, we had to deal with obstacles in the sites. In order to avoid obstacles, we used the strategy explained in Chapter 6: we used some additional points (with no production) to describe the obstacles and give flexibility to the routes. Let us consider the West site first. In order to understand where to better locate these extra points, we first optimized the route without considering any obstacle. The result is shown in Figure 10.7: point 0 (in red) is the substation, where we have to collect the energy; points 1-8 describe the existing obstacle; points 9-32 are turbines that need to be connected to the substation. The two available cable types are plotted with two different colours in the figure.

We now included the obstacle in the optimization by connecting the extra-points (1-8) with zero-cost cables (in black): due to the no-crossing constraints, cables cannot now pass through the obstacle. We also used additional fake points around the obstacle to increase flexibility in the route. In this case the optimizer reached optimality in a matter of seconds on a standard PC. The resulting cable route is in Figure 10.8.

The same process was done for the East site. In this case we had multiple obstacles: two circular obstacles inside the site, and the export cable on the west border of the site (that could not be crossed). They are indicated with black lines in Figure 10.9. We used the same cable types as before, imposing

Figure 10.7: Optimal cable routing for the West site, without considering the obstacle (points 1-8)

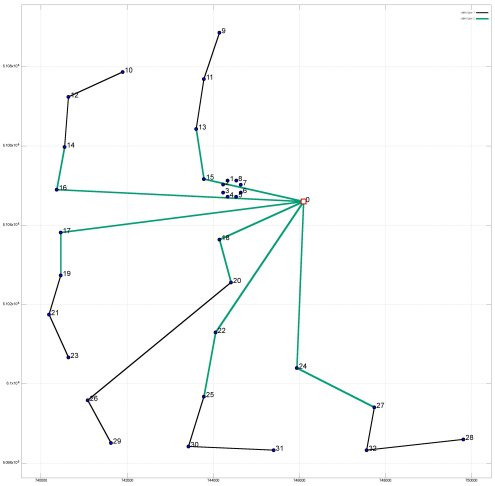
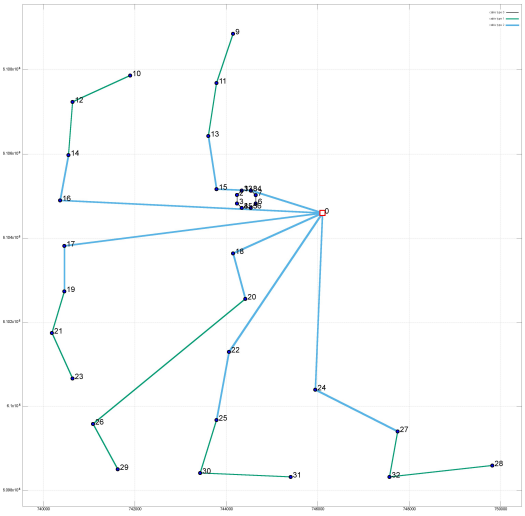


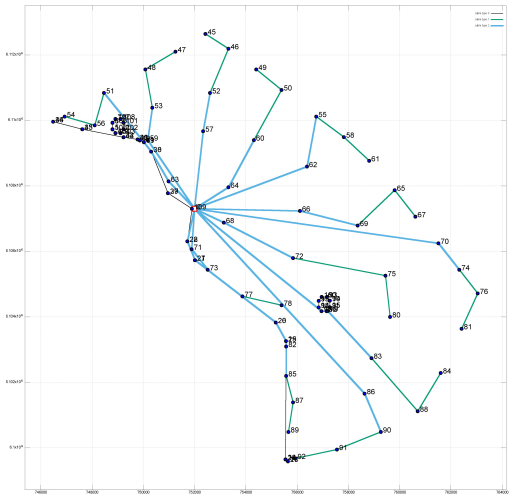
Figure 10.8: Optimal cable routing for the West site, considering the obstacle (black circle)



a string structure and maximum of 12 strings connected to the substation. This instance was more challenging for the optimizer, due to a larger number of turbines and the more challenging constraints (in particular, in terms of

obstacles). The optimizer reached the timelimit of 2 hours with a 1.8% gap. Figure 10.9 shows the resulting cable routing.

Figure 10.9: Optimal cable routing for the East site, considering obstacles (in black)



These outputs from our software are post-processed to define the business case (and LEC evaluation) for the whole project. In particular, the production revenue and the costs estimated by our optimizers are recomputed with more refined functions.

The value of the entire project is evaluated considering price fluctuations, costs of the different components, expected turbine downtime due to failures, construction costs, maintenance costs and so on. Different scenarios are tested and compared (e.g., considering different turbine types on the market, or different cable manufacturers) until the best business case is defined.

10.3 Conclusions

This chapter shows in more details how the optimizers are used inside the company, following more closely one scenario evaluation for the Danish Kriegers Flak project. Before getting to the final layout, this process is iterated multiple times in order to fully understand the impact of different design choices. According to Vattenfall's experts, this process would have been impossible without optimizers.

Acknowledgments

We would like to thank Mads Krogsgaard who followed the Danish Kriegers Flak case using our optimizers and developed models for its business case evaluation.

Part V

A bit of Machine Learning

Chapter 11

Machine Learning meets Mathematical Optimization to predict the optimal production of offshore wind parks

Martina Fischetti[a] · Marco Fraccaro[b]

[a]Vattenfall and Technical University of Denmark, Department of Management Engineering, Produktionstorvet, Building 424, DK-2800 Kgs. Lyngby, Denmark

[b]Technical University of Denmark, DTU Compute, Department of Applied Mathematics and Computer Science, Richard Petersens Plads 321, 2800 Kgs. Lyngby, Denmark

Publication Status: Published as Fischetti and Fraccaro, 2018

Reading Instructions: Usage of our model Fischetti and Monaci, 2016a together with Machine Leaning techniques to predict optimized production. If you read Chapter 4, you can skip Section 11.3

Abstract: In this paper we propose a combination of Mathematical Optimization and Machine Learning to estimate the value of optimized solutions. In particular, we investigate if a machine, trained on a large number of optimized solutions, could accurately estimate the value of the optimized solution for new instances. In this paper we will focus on a specific application: the offshore wind farm layout optimization problem. Mixed Integer Programming models and other state-of-the-art optimization techniques, have been developed to solve this problem. Given the complexity of the problem and the big difference in production between optimized/non optimized solutions, it is not trivial to understand the potential value of a new site without running a complete optimization. This could be too time consuming if a lot of sites need to be evaluated, therefore we propose to use Machine Learning to quickly estimate the potential of new sites (i.e., to estimate the optimized production of a site without explicitly running the optimization). To do so, we trained and tested different Machine Learning models on a dataset of 3000+ optimized layouts found by the optimizer. Thanks to the close collaboration with a leading company in the energy sector, our model was trained on real-world data. Our results shows that Machine Learning is able to efficiently estimate the value of optimized instances for the offshore wind farm layout problem.

11.1 Introduction

Mathematical Optimization (MO) and Machine Learning (ML) are two closely related disciplines that have been combined in different way. A very popular application of the two together is the so-called Prescriptive Analytics field Bertsimas and Kallus, 2014, where ML is used to predict a phenomenon in the future, and MO techniques are used to optimize an objective over that prediction. A popular research area is now also to use Machine Learning to improve heuristics decisions in Mixed Integer Linear Programming (MILP) algorithms, for example in the branching procedure Lodi and Zarpellon, 2017; Alvarez et al., 2017; Khalil et al., 2016 or in decomposition techniques Kruber and Marco, 2017.

The paper by Bello et al., 2016 introduces a framework to tackle combinatorial optimization problems using neural networks and reinforcement learning; the authors study the traveling salesman problem and train a recurrent network to predict a probability distribution over different solutions. In Dai et al., 2017 the authors investigate instead how to learn a heuristic algorithm by exploiting the structure of the instances of interest; the resulting approach is applied to different optimization problems over graphs, namely the minimum vertex cover, the maximum cut and the traveling salesman problems. The work of Hottung et al., 2017 integrates ML and tree search to derive a truncated branch-and-bound algorithm using bound estimates; the proposed method, called deep-learning assisted heuristic tree search, is applied to the so-called container pre-marshalling problem. The problem of deciding at which node of a branch-and-bound tree a heuristic should be run is instead addressed in Khalil et al., 2017, where ML is used to predict whether a certain heuristic would improve the incumbent if applied at a given branching node. Finally, the very recent special issue "Combining Constraint Solving with Mining and Learning" 2017; Passerini et al., 2017 contains a collection of papers that combine ML with constraint programming/optimization. In particular, the work of Berg and Järvisalo, 2017 studies the applicability of Boolean optimization (maxSAT) to clustering problems, while Bartlett and Cussens, 2017 uses integer linear programming for learning Bayesian network structures, and Bessiere et al., 2017 investigates an architecture for acquiring constraint programming constraints from classified examples.

In the present work, we will instead investigate a different way to merge MO and ML, where the slow MO model comes first, and its (almost) optimal solutions are used as training set for a Machine Learning (ML) algorithm that can quickly estimate the value of new optimized solutions. We then show that a ML model can leverage data collected by running state-of-the-art MO algorithms on a specific application to learn to predict how MO would perform in new instances of the same task, without the need to run the computationally expensive MO algorithm every time. In this work, we will focus on a specific application, already studied by the first author in Fischetti and Monaci, 2016b, namely the offshore wind park layout optimization problem. This idea can however be used for many different optimization problems and applications, such as transport, logistics and scheduling.

The *wind farm layout optimization problem* consists in finding an optimal allocation of turbines in a given site, to maximize the park power production. A particularly challenging feature of this problem is the interaction between turbines, also known as *wake effect*. The wake effect is the interference phenomenon for which, if two turbines are located one close to another, the upwind

one creates a shadow on the one behind (see Figure 11.1). This is of great importance in the design of the layout since it results into a loss of power production for the turbine downstream, that is also subject to a possibly strong turbulence. For many years, this issue has been unknown or underestimated, and old wind parks have been designed with a very regular (and highly wake-affected) layout. Many of the parks operating today are on a regular layout. It was estimated in Barthelmie et al., 2009 that, for large offshore wind farms, the average power loss due to turbine wakes is around 10-20% of the total energy production.

Figure 11.1: Wake effect on Horns Rev 1 wind park. Many offshore wind parks operating today are still on a regular grid, and so largely affected by wake effect. Source: Vattenfall, 2017

It is then obvious that power production can increase significantly if the wind farm layout is properly optimized. The large size of the problem, the complexity of the wake effect and the presence of other constraints, makes it impossible to create a good layout without the usage of an advanced optimization tool. Since the difference in power production between optimized solutions and unoptimized ones can be significant, it is even difficult to estimate the potential power production of a site, without running a complete optimization of the layout.

In this paper, we want to use a ML algorithm able to better estimate the potential of a site, without running a complete optimization. The MILP-based approach proposed in (Fischetti and Monaci, 2016b), takes about 10 hours for a complete optimization. Acceptable preliminary solutions, used for what-if analyses, may be found in 1 hour. Once a company has decided to invest on a site, this amount of time is not an issue, and the MO tools are used (1-hour runs for initial what-if analyses, and 10-hour runs for the final layouts). Nevertheless, there are other cases in which even a 1-hour optimization is not an option,

due to the very large number of options to be compared. This happens, for example, when the site where to build the park is not decided a priori. Indeed, in the future it may be up to the constructing company to select the site where to build the park. The company would then look for all the zones in the world with good wind conditions, and it will have to choose between a large amount of areas. Just assuming that 1000+ potential sites may be identified, running (even a fast 1-hour) optimization algorithm for all of them would be infeasible. A properly trained ML algorithm could instead be able to identify the most promising sites in a matter of minutes. Once the site is selected, a detailed (and more time-consuming) optimization can be run to define the actual layout. In this work we will focus on the setting where a lot of sites are available and we aim at having a fast ML algorithm to pre-select the most promising ones, and then run a full optimization on those only. A further motivation for using ML for pre-selection is as follows. In the near future, the turbine manufacturer may design a site-specific turbine, hence thousands variants of a turbine should be evaluated to decide which one better fits the site of interest. Also in this case, running even a fast 1-hour optimization is not an option, and ML can instead be used to pre-select the best turbine-models. Having in mind the site-selection application, we address the case where a company wants to construct a specific number of turbines in an offshore area. Even if the production would increase by spreading the turbines due to the wake effect, the infrastructure costs to connect turbines very far away from each other would also increase. Therefore we assume that, even if a huge offshore area is available, the company would discretize it in a number of smaller rectangular sites. These sites can have different dimensions and the wind can largely vary from site to site. The potential of a site depends not only from the site itself but also on the turbine used. The company could also investigate the potential of different sites for different turbine types. We therefore considered rectangular instances of different dimensions, with different wind scenarios (taken from real-world parks) and with different turbine types. In this study we considered standard turbines already available on the market. We defined and optimized over 3000 instances using the MO tool developed in Fischetti and Monaci, 2016b. The power production of the heuristic solutions found is used as training set for our ML algorithm.
A distinguished feature of our work is that we do not expect to estimate the optimal *solution* (which is arguably very problematic for the state of the art), but we content ourselves with the estimate of the optimal *value* of it.

The paper is organized as follows. Section 11.2 gives an overview of basic energy concepts that can be useful to the reader to better understand the wind farm problem. Section 11.3 summarizes the MILP model used in the optimization phase. Section 11.4 defines the input instances used and studies

the feature we will use for our ML model. Section 11.5 describes the alternative ML models we considered, namely Linear Regression, Neural Networks and Support Vector Regression. These models are tested on a large dataset and compared in Section 11.6. Finally Section 11.7 draws some conclusions and discusses possible extensions.

11.2 Basic wind energy notions

In this section we give a basic introduction to wind energy concepts that can help the reader to better understand the problem in hand. Our work can be valuable, indeed, both as a proof of concept on the possibility of estimating the value of optimized solutions using ML techniques, and from a practical application in the wind sector. For the reader interested in the second topic, we offer in this section a fast review of useful wind energy concepts. For further details, see Fischetti, 2014.

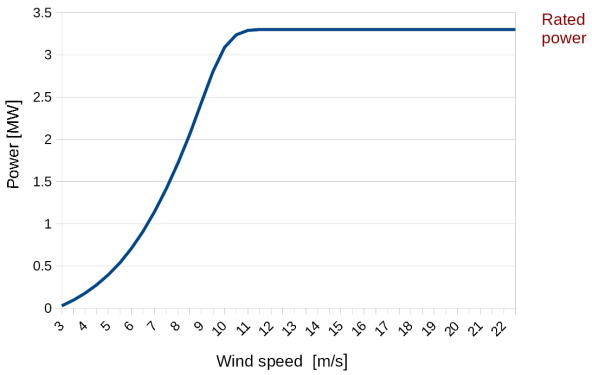
The wind energy sector is a fast growing and more and more competitive sector. In order to build more profitable parks, it is very important to reduce costs and to increase incomes (i.e. increase power production). Nowadays, it is normally up to the State to define a specific site to construct a new offshore park, and different constructing/operating companies will enter a tender to have the rights to construct it. The company that can offer energy at the lowest cost will win the tender. A key feature to lower the KW/h price is to produce more energy at the same cost, and therefore the optimization of the layout (in order to reduce wake effect) is a key factor. In this context, MO techniques are used by many energy companies. Even if these techniques could require hours of computation, this is not a limitation, since the number of runs is quite limited. In this paper, instead, we look at a different context where we suppose that is up to the constructing/operating company to decide where to build the park. Supposing to have maximum freedom in this choice, the possible sites in a very large offshore area would be too many to run a complete optimization for each of them. In this context the company would be interested in a first (fast) evaluation of the sites, in order to rank the best ones, and focus its attention on those. The way this fast (preliminary) evaluation of sites is done today is by evaluating the production of a regular layout, obtained by positioning turbines on a grid. It is nevertheless well known in the wind sector, that regular layouts are highly wake-affected, hence this estimate (as we will see later) fails at capturing the potential of new sites.

11.2.1 Wind turbines

Once a company has decided a site, it is a key factor to decide which kind of turbines to build in the site.

A wind turbine model is typically identified by its manufacturer (for example, Siemens, Vestas, Adwen etc.), by its dimensions and its power rating. The power rating is a measure of how powerful a turbine is. All turbines produce different power at different wind speeds: the higher the wind, the higher the production. A typical power curve is shown in Figure 11.2: the power production increases with wind and it saturates at a certain MW production. This value is the *rated power* of a turbine. Another interesting turbine specification is its rotor diameter, that identifies the area spanned by the turbine blades (see Figure 11.3). In general, the bigger the rotor, the higher the rated power of a turbine. The manufacturer can also design turbines with the same rated power but different rotor diameters. In this case, the difference mainly occurs at medium wind speeds (around 5 – 15 m/s), when a bigger rotor is able to catch more wind and produce more, even if at higher wind speeds the turbines are controlled to not produce more than the rated power. In Figure 11.4 we plot the power curve of two turbine models with the same rated power but different rotor diameters. It is therefore important, in the specification of a turbine model, to indicate also its rotor diameter.

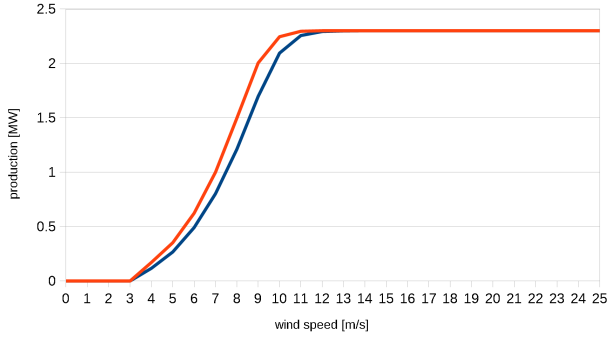
Figure 11.2: Example of power curve



It is then clear that, if we decide to build a fixed number of turbines, the turbine model selected will impact the total power production of the site. This is why, as we will see later, it is important to capture this informations both in the

Figure 11.3: Rotor diameter of a turbine. Picture inspired by an ollustration from Vattenfall, 2017

Figure 11.4: Example of power curve of two turbines with the same rated power but different diameters (101m in blue and 113m in red).



MO and in the ML models.

11.2.2 Site and wind farm production

In this paper we are considering the offshore wind farm design problem. The term *offshore* refers to the construction of wind farms in bodies of water, so the site is normally a regular area in a sea zone. This has some advantages compared to onshore sites, for example the wind is generally stronger and equally distributed. In addition, residents opposition to construction is usually much weaker for offshore parks, and the so-called NIMBY ("Not In My BackYard") problem is easier to solve. In the offshore case, the height of the turbines above sea level is supposed to be equal everywhere, so the offshore case is actually a 2D case. This also means that, in offshore sites, we can consider the wind to be equally distributed all over the site, so the potential power production of all possible positions is the same.

A key role is then played by the interference between turbines, that causes a loss of production in some positions. In this paper we used Jensen's model Jensen, 1983 to compute the interference caused by a turbine: the interference is modelled as a cone centered in the upwind turbine. The further away we go from the upwind turbine, the lower the interference. Figure 11.5 plots the interference cones for a wind blowing from South-East. In the plot, red dots represent built turbines, while the background colours represent the power loss [MW] due to interference (a turbine placed in a dark blue area, will not suffer any wake effect). Looking at this plot it is easy to imagine that regular layouts suffer from higher interference, since turbines often lay in the interference cone of other turbines.

The wake effect, of course, depends on wind intensity and wind direction. Nevertheless, once the position of the turbines (i.e., the layout of the park) is selected, the turbines cannot be moved depending on the wind. Therefore, when we optimize the turbine position, we have to consider all the possible wind scenarios with their probability (as we will more formally see in Section 11.3). Figure 11.6 shows the wake effect values in a site, when considering all its different wind scenarios.

11.3 The optimization model

The aim of this section is to outline a MILP model for the offshore wind farm layout problem. Additional constraints and sophisticated algorithms to solve large scale instances have been developed by the first author (see Fischetti and

Figure 11.5: Wake effect. The red dots represent built turbines, the background colors represent the interference intensity according to Jensen's model (wind blowing from south-est)

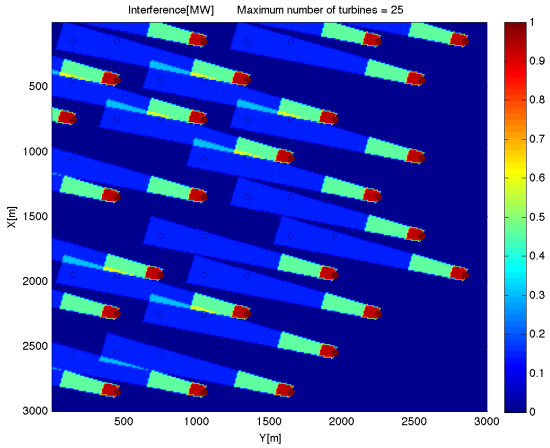
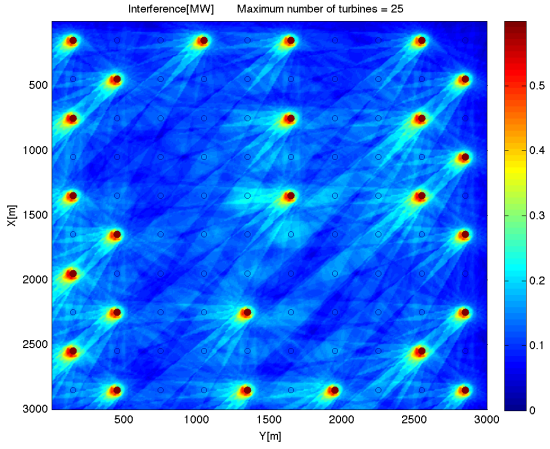


Figure 11.6: Wake effect. The red dots represent build turbines, the background colors represent the interference intensity according to Jensen's model (when considering wind statistics from a real-world park)



Monaci, 2016b), but their description is out of the scope of the present paper.

At the optimization stage, an offshore site is given together with the wind statis-

tics in the site. We are asked to determine the optimal allocation of turbines in the area, in order to maximize the park power production. The optimization needs to consider that a minimum and maximum number of turbines can be built, a minimal separation distance must be guaranteed between two turbines to ensure that the blades do not physically clash (turbine distance constraints), and the power loss due to wake effect is minimized.

We first discretize the area in a number of possible turbine positions. Let V denote this set and let

- P_i be the power that a turbine would produce if built (alone) at position i.
- N_{MIN} and N_{MAX} be the minimum and maximum number of turbines that can be built in the park, respectively;
- D_{MIN} be the minimum distance between two turbines;
- $dist(i,j)$ be the symmetric distance between positions i and j.
- I_{ij} be the interference (loss of power) experienced at position j when a turbine is installed at position i, with $I_{jj} = 0$ for all $j \in V$. We used a Jensen's model to compute it Jensen, 1983;

In addition, let $G_I = (V, E_I)$ denote the incompatibility graph with $E_I = \{[i,j] : i,j \in V,\, dist(i,j) < D_{MIN},\, j > i\}$.
Note that the interference matrix I is not symmetric, as the loss of power due to interference experienced by i when a turbine is installed in site j depends on the relative position of i with respect to j but also on their relative position with respect to the wind direction.

In our model, we define binary variables x_i for each $i \in V$ as

$$x_i = \begin{cases} 1 & \text{if a turbine is built at position } i \in V; \\ 0 & \text{otherwise.} \end{cases}$$

The quadratic objective function (to be maximized) reads $\sum_{i\in V} P_i x_i - \sum_{i\in V}(\sum_{j\in V} I_{ij}x_j)\, x_i$ and can be restated as

$$\sum_{i\in V}(P_i x_i - w_i) \tag{11.1}$$

where

$$w_i := \big(\sum_{j\in V} I_{ij}x_j\big)\, x_i = \begin{cases} \sum_{j\in V} I_{ij}x_j & \text{if } x_i = 1; \\ 0 & \text{if } x_i = 0 \end{cases}$$

denotes the total interference caused by position i. Our MILP model then reads

$$\begin{aligned}
\max \quad & z = \textstyle\sum_{i \in V} (P_i x_i - w_i) && && (11.2)\\
\text{s.t.} \quad & N_{MIN} \leq \textstyle\sum_{i \in V} x_i \leq N_{MAX} && && (11.3)\\
& x_i + x_j \leq 1, && \forall \{i,j\} \in E_I && (11.4)\\
& \textstyle\sum_{j \in V} I_{ij} x_j \leq w_i + M_i (1 - x_i), && \forall i \in V && (11.5)\\
& x_i \in \{0,1\}, && \forall i \in V && (11.6)\\
& w_i \geq 0, && \forall i \in V. && (11.7)
\end{aligned}$$

where the big-M terms $M_i = \sum_{j \in V : [i,j] \notin E_I} I_{ij}$ are used to deactivate constraint (11.5) in case $x_i = 0$. This model works on larger instances compared with equivalent models in the literature involving 2-index variables $y_{ij} = x_i x_j$; see Fischetti, 2014 for details. Note that constraint (11.3) can be used to impose the construction of a fixed number of turbines by setting $N_{MIN} = N_{MAX}$.
Of course, the power production P_i and the interference value I_{ij} vary with the wind. Using statistical data, one can in fact collect a large number, say K, of wind scenarios k, each associated with its own P_i^k, I_{ij}^k and with a probability π_k. As shown in Fischetti and Monaci, 2016b, one can then take wind scenarios into account by simply defining $P_i := \sum_{k=1}^{K} \pi_k P_i^k \;\; (i \in V)$ and $I_{ij} := \sum_{k=1}^{K} \pi_k I_{ij}^k \;\; (i,j \in V)$.

To solve large-scale instances (with 20 000+ possible positions) some ad-hoc heuristics and a MILP-based proximity search Fischetti and Monaci, 2014 heuristic have been used on top of this basic model. We refer the interested reader to Fischetti and Monaci, 2016b for details.

11.4 Data Generation

We used the model presented in Section 11.3 to determine the optimized power production of a large number of realistic instances. These instances have been artificially created by considering rectangular areas of different sizes, different real-world turbine types, and different wind statistics from different real-world sites. In particular, we generated different sites by generating sets of possible

points on a regular grid (10m point-to-point distance) inside rectangles of different dimensions (all possible combinations of edge sizes 6000, 7000, 8000, 9000, 10000, 11000, 12000, 13000 and 14000m). We computed power production and interference based on the data from the following real-world turbines:

- Adwen 8 MW, with a rotor diameter of 180m;
- Vestas 8.4 MW, with a rotor diameter of 164m;
- Siemens 7 MW, with a rotor diameter of 154m;
- Vestas 8 MW, with a rotor diameter of 164m;
- Siemens 3.2 MW, with a rotor diameter of 113m;
- Siemens 2.3 MW, with a rotor diameter of 101m.

For each line of the list above we have specified the turbine manufacturer (Adwen, Vestas, Siemens), the rated power of the specific model (i.e. the maximum MW power that the turbine can produce), and the turbine rotor diameter. Note that different rated powers and different rotor diameters affect the power production P_i and the interference $I_{i,j}$ of each turbine and therefore the total power production of the park. Finally, we considered different real-world wind statistics for the wind scenarios, namely those from the real offshore wind parks named Borssele 1 and 2, Borssele 3 and 4, Denish Krigers Flak, Ormonde, Hollandse Kust Zuid and Horns Rev 3. These are in-operation or under-construction parks located in The Netherlands, Denmark, and United Kingdom. By considering all the possible combinations of sites, turbine types and winds, we obtained about 3000 instances. We imposed that a fixed number of 50 turbines needs to be located in the site. For each instance we computed:

1. Grid production: the power production of a solution obtained by locating the turbines on a regular grid; and
2. True production: the (almost) optimal power production computed by our MO tool.

We decided to compute the production of a layout on a regular grid for two reasons. First, this represents how a manual operator would evaluate the potential of a site, and so it acts as a benchmark value for our ML models. Secondly, this value is very informative: as we will better see in Section 11.4.1, it captures the dimension and the wind of the site, while providing a lower bound for our optimized solution.

It is important to observe that the grid production requires short computing time and can be calculated in a pre-processing step. Optimization for the difficult case (true production) was instead obtained through the MILP-based heuristic of Fischetti and Monaci, 2016b, with a time limit of 1 hour on a standard PC using IBM ILOG CPLEX 12.6.

The output of the optimization has been used to train the ML models presented in Section 11.5, where the grid production is considered an input feature, while the true production is the figure that we aim at estimating (hence its name).

11.4.1 Feature definition

In order for our ML models to capture the wind park problem, it is very important to describe its characteristics in a meaningful way. In particular, we need to give to the ML models valuable information on the turbine type used, on the wind and on the site. In order to asses which are the most useful information to include we used our knowledge of the problem and different visualizations of the data. As a result of our analysis, the following features have been selected:

- the rated power of the turbine: this is the maximum power (MW) that the turbine can produce (at high wind speeds); it describes the turbine model, and impacts both park production and interference;
- the rotor diameter of the turbine: this describes the dimensions of the turbine; it impacts the production, the interference and the minimum distance between turbines;
- the square root of the area of the site (expressed in rotor diameters);
- the ratio between the two edges of the rectangle: this captures the shape of the site;
- the production of a regular layout on a grid: this is the output of case 1) in Section 11.4, and captures both the wind in the site and the site dimensions (as the bigger the site, the lower the interference).

Figure 11.7 plots the relation between our selected features and the optimized production we aim at estimating. It can be noticed, in particular, that the production increases having bigger sites, since the optimizer can better spread the turbines (Figure 11.7a). For an easier reading of our plot we here indicate

the square root of the area in m (while we express it in rotor diameters when we give it on input to the ML models). Analysing our data we can notice that production also depends on the shape of the site: squared sites have slightly higher production (Figure 11.7b). Since in our problem we suppose to build a fixed number of turbines (50) at, at least, 5 rotor diameters of distance, the optimized production highly depends also on the selected turbine model: the more powerful (bigger) the turbine, the higher the production (Figures 11.7c and 11.7d). Finally, we can see in Figure 11.7e that the production of a regular layout is highly correlated to the optimized production: if we have a big site with good wind and powerful turbines, both the the grid production and the optimized one will be high. Nevertheless, the relation is not linear as there are some sites that are underestimated by the production on a grid: in Figure 11.7e, one can notice that multiple sites are evaluated as equivalent by the production on a grid, but not by the optimizer. This is the case, for example, for sites with grid production 185 (vertical dotted line in Figure 11.7e) that have different optimized productions.

In order for our ML models to work, we need to encode our features in a way that is easy for the model to interpret. This is why, for example, we preferred to use the ratio between edges instead of their length. In the same way, we do not explicitly pass the wind statistics of the site to the ML model, but we prefer to use the regular-grid production (that contains a much richer information, since it relates already the wind with the site dimensions and the turbines used).

11.5 Machine Learning

We defined three different ML models to estimate the reduction in power production due to the interference, namely Linear Regression, Neural Networks (NNs) and Support Vector Regression (SVR). In order to benchmark our solutions we used the production on a regular grid layout as a baseline. This is, indeed, the approach that practitioners would use to quickly estimate the potential of a site.

The features described in the previous session are provided on input to the ML models through a vector $\mathbf{x}$. The estimated optimized power production $\hat{y}$ is then modelled through a function f that depends on some unknown parameters $\mathbf{w}$ learned during the training phase, i.e.

$$\hat{y} = f(\mathbf{w}, \mathbf{x}) \ .$$

We used the *Root Mean Squared Error (RMSE)* Bishop, 2006 to measure the

quality of our ML models. RMSE is a widely-used measure in regression models, as it gives a good description of the deviation between the predictions of the model and the "true" values. Given a training data set containing input-output pairs $\{(\mathbf{x}_n, y_n), n = 1, \ldots, N_{train}\}$, the RMSE formula reads:

$$E(\mathbf{w}) = \sqrt{\frac{1}{N_{train}} \sum_{n=1}^{N_{train}} (y_n - f(\mathbf{w}, \mathbf{x}_n))^2}$$

where $E(\mathbf{w})$ is minimized when our estimate $f(\mathbf{w}, \mathbf{x}_n)$ is as close as possible to the "real value" y_n. The optimal parameters $\mathbf{w}^\star$ for our ML models are therefore found as $\mathbf{w}^\star = \arg\min_{\mathbf{w}} E(\mathbf{w})$. For Linear Regression this minimization problem is solved easily, as it is a convex problem with an analytic solution. The drawback of linear regression, however, is that it can only model linear dependencies in the data. We therefore defined more powerful models by introducing some non-linearities, in the form of stacked layers of non-linear functions for the *NN* model, or kernels for the SVR. A detailed description of these models and their optimization procedures is out of the scope of this work. The interested reader is referred to Bishop, 2006 for an in-depth discussion on the topic.

11.6 Results

We divided our 3000+ instances in training and test set, constructed to reflect the practical application of this work: the company would use our ML model to evaluate the potential of new sites (not seen during training), where the main difference is in the wind statistics. The training set consist of 2268 instances, corresponding to the winds statistics of the real sites Borssele 1 and 2, Borssele 3 and 4, Horns Rev 3, and Denish Krigers Flak. These instances are used to train and calibrate our ML models. The remaining 1134 instances, that correspond to the wind statistics of Ormonde and Hollandse Kust Zuid, are used as test set to evaluate the performances of the different ML models, and will give us a measure of how much our models *generalize* to previously-unseen data.

As these models are very sensitive to different scaling of the input features, we standardized all the features to have mean 0 and standard deviation 1 over the training set. The hyperparameters of the models (e.g., the number of layers and units of the NN or the kernel type in the SVR) are chosen using the *scikit-learn* Pedregosa et al., 2011 function *GridSearchCV*, that exhaustively considers all parameter combinations on a grid (5-fold cross-validated on the training set).

Table 11.1: RMSE of the test set for the different models.

Model	RMSE
Grid	4.15
Linear Regression	1.15
Neural Network	2.24
Support Vector Regression	0.74

According to our tests, the best NN is a single-layer network with 20 hidden units and hyperbolic tangent non-linearity. For the SVR, the best performing kernel function is the Radial Basis Function (Gaussian) kernel. All the models were again implemented using Python's machine learning library *scikit-learn*.

In Table 11.1 we compare the performance of the models in terms of RMSE on the test set. We see that all models outperform by a large margin the baseline given by the production on a regular grid (denoted as *Grid* in the table), with Support Vector Regression being the best among them. Importantly, all the ML models can be evaluated on new instances in less than a second, as opposed to the 1-hour runs needed when using the MO algorithm.

In Figure 11.8 we visualize the predictions of the models on the test set: on the y-axis we report the true optimized production, and on the x-axis its estimate from the model (each point in the graph represents a different test instance). If the predictions were perfect, all the points would lay on the $y = x$ line (in red in the plot).

An alternative view of the predictions from the model can be found in Figure 11.9 that shows, for some of the instances, the *predicted* optimal production from the different ML models, the one obtained with the *grid* solution (what a practitioner would estimate) and the optimized (*true*) production. From the different plots it is clear the importance of both MO and ML in this problem. The optimization of the layout is extremely valuable, since it can increase the production of the park (as the difference between the green and the orange line shows). In order to fast and accurately evaluate the potential of a new site, ML models can provide accurate prediction. In particular SVR models highly outperform the manual operator strategy (grid) and are able to predict very closely the optimized production value.

As we shortly discussed already by commenting Figure 11.7e, it is very important to use MO+ML optimization to evaluate a site rather than only grid production, as the grid production may result in a wrong ranking of the sites.

In Figure 11.10 we show the difference between the true production and the grid production, and the difference between true and the predicted production (for SVR). From the plot one can notice that the difference between the real and the predicted production is always very low (around 0), while the grid tends to underestimate the sites in almost all cases. The difference between the true and the grid production varies a lot from instance to instance: this makes it impossible to simply improve the grid estimate (for example, using a scaling factor), and proves the value of more complex ML techniques.

11.7 Conclusions and future work

In the present work we showed the relevance of using MO and ML techniques together. We have shown that ML techniques (SVRs in particular), trained on a large number of (almost) optimal solutions, can predict the optimal value of new instances of the same problem rather well. In this work, we have focused on the wind park layout problem. The exact value of optimized solutions for this specific problem is very difficult to compute, given the large number of constraints and non-linearities involved, and can only be obtained using a sound MO tool. In our tests we have shown that the ML estimate highly outperforms the human one.

A possible extension of the model can be to allow for different numbers of turbines or different shapes of the park site (not only rectangles). More ambitiously, future work could investigate the application of our approach to different OR problems. One could, indeed, address the problem of estimating the optimal *value* of an optimization problem by using ML algorithms trained on optimized solutions computed off-line by time-consuming MO solvers. This estimate can be of interest by itself (as in the wind farm application studied in this paper), but can also be very useful, e.g., for heuristic node pruning in a branch-and-bound solution scheme.

References

Alvarez, A. M., Q. Louveaux, and L. Wehenkel (2017). "A Supervised Machine Learning Approach to Variable Branching in Branch-And-Bound". In: *INFORMS Journal on Computing* 29.1, pp. 185–195.

Barthelmie, R., K. Hansen, S. T. Frandsen, O. Rathmann, J. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. Politis, and P. K. Chaviaropoulos (2009). "Modelling and measuring flow and wind turbine wakes in large wind farms offshore". In: *Wind Energy* 12, pp. 431–444.

Bartlett, M. and J. Cussens (2017). "Integer Linear Programming for the Bayesian network structure learning problem". In: *Artificial Intelligence* 244. Combining Constraint Solving with Mining and Learning, pp. 258–271. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2015.03.003. URL: https://www.sciencedirect.com/science/article/pii/S0004370215000417.

Bello, I., H. Pham, Q. V. Le, M. Norouzi, and S. Bengio (2016). "Neural Combinatorial Optimization with Reinforcement Learning". In: *CoRR* abs/1611.09940. arXiv: 1611.09940. URL: http://arxiv.org/abs/1611.09940.

Berg, J. and M. Järvisalo (2017). "Cost-optimal constrained correlation clustering via weighted partial Maximum Satisfiability". In: *Artificial Intelligence* 244. Combining Constraint Solving with Mining and Learning, pp. 110–142. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2015.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0004370215001022.

Bertsimas, D. and N. Kallus (2014). "From Predictive to Prescriptive Analytics". In: *eprint arXiv:1402.5481*.

Bessiere, C., F. Koriche, N. Lazaar, and B. O'Sullivan (2017). "Constraint acquisition". In: *Artificial Intelligence* 244. Combining Constraint Solving with Mining and Learning, pp. 315–342. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2015.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0004370215001162.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Dai, H., E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song (2017). "Learning Combinatorial Optimization Algorithms over Graphs". In: *CoRR* abs/1704.01665. arXiv: 1704.01665. URL: http://arxiv.org/abs/1704.01665.

Fischetti, M. (2014). "Mixed-Integer Models and Algorithms for Wind Farm Layout Optimization". http://tesi.cab.unipd.it/45458/1/tesi_Fischetti.pdf. MA thesis. University of Padova.

Fischetti, M. and M. Monaci (2014). "Proximity search for 0-1 mixed-integer convex programming". In: *Journal of Heuristics* 6.20, pp. 709–731.

Fischetti, M. and M. Fraccaro (2018). "Machine Learning meets Mathematical Optimization to predict the optimal production of offshore wind parks". In: *Computers and Operations Reserach*. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2018.04.006.

Fischetti, M. and M. Monaci (2016a). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: 10.1007/s10732-015-9283-4. URL: https://doi.org/10.1007/s10732-015-9283-4.

Fischetti, M. and M. Monaci (2016b). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397.

Hottung, A., S. Tanaka, and K. Tierney (2017). "Deep Learning Assisted Heuristic Tree Search for the Container Pre-marshalling Problem". In: *CoRR* abs/1709.09972. arXiv: 1709.09972. URL: http://arxiv.org/abs/1709.09972.

Jensen, N. (1983). *A note on wind generator interaction.* Tech. rep. Technical Report Riso-M-2411(EN), Riso National Laboratory, Roskilde, Denmark.

Khalil, E. B., P. L. Bodic, L. Song, G. Nemhauser, and B. Dilkina (2016). "Learning to Branch in Mixed Integer Programming". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16) Learning.*

Khalil, E. B., B. Dilkina, G. L. Nemhauser, and Y. S. Shabbir Ahmed (2017). "Learning to Run Heuristics in Tree Search". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 659–666. DOI: 10.24963/ijcai.2017/92. URL: https://doi.org/10.24963/ijcai.2017/92.

Kruber, M. and E. L. Marco (2017). "Learning when to use a decomposition". In: *Proceeding of CPAIOR2017.*

Lodi, A. and G. Zarpellon (2017). "On learning and branching : a survey". In: Technical Report, pp. 1–29.

Passerini, A., G. Tack, and T. Guns (2017). "Introduction to the special issue on Combining Constraint Solving with Mining and Learning". In: *Artificial Intelligence* 244. Combining Constraint Solving with Mining and Learning, pp. 1–5. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2017.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0004370217300036.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

"Combining Constraint Solving with Mining and Learning" (2017). In: *Artificial Intelligence* 244, IFC–. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(17)30008-5. URL: https://www.sciencedirect.com/science/article/pii/S0004370217300085.

Vattenfall (2017). *internal image storage.*

Figure 11.7: Relation between optimal power production and our input features.

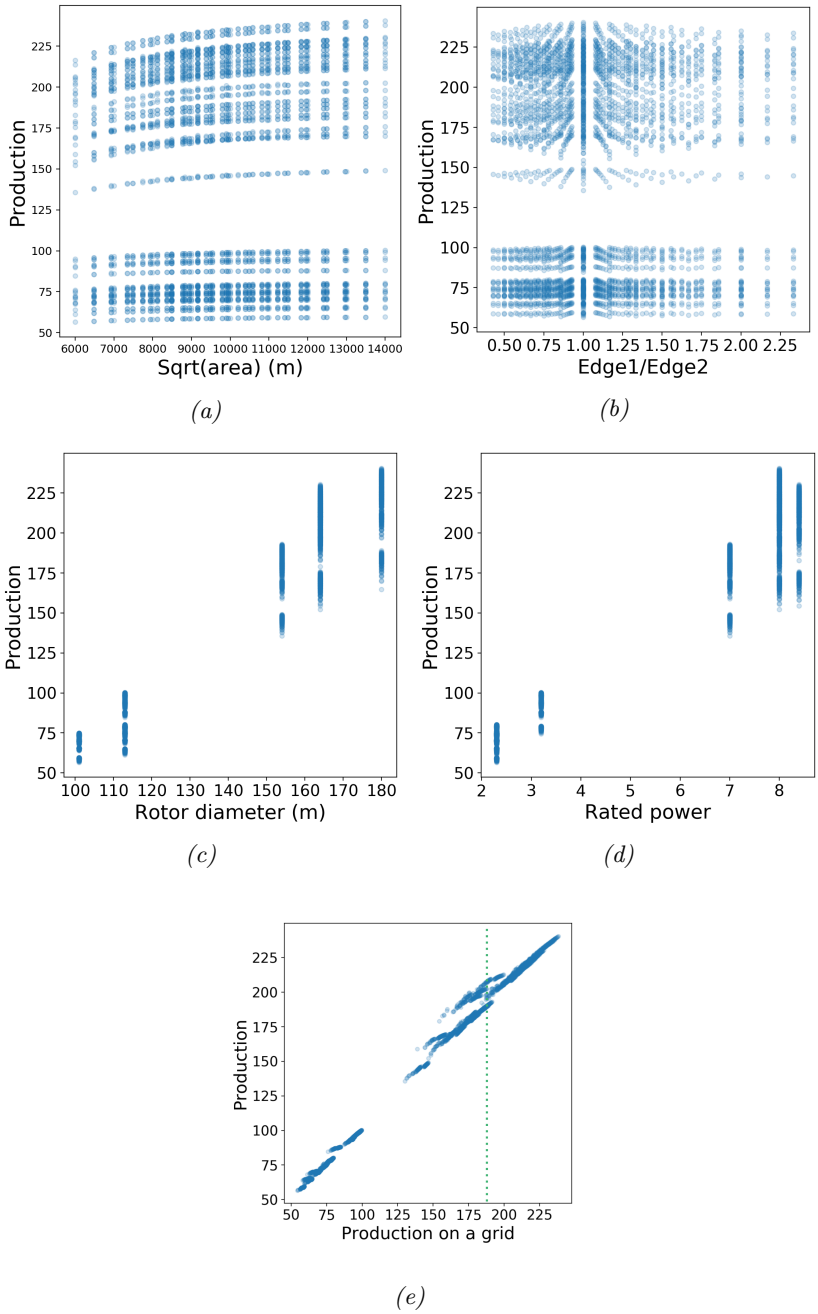
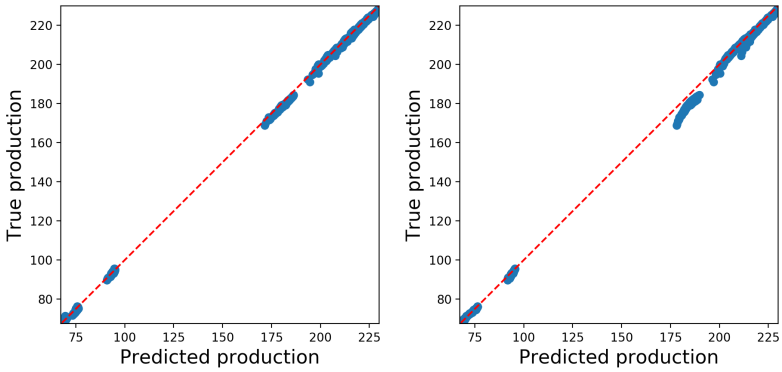
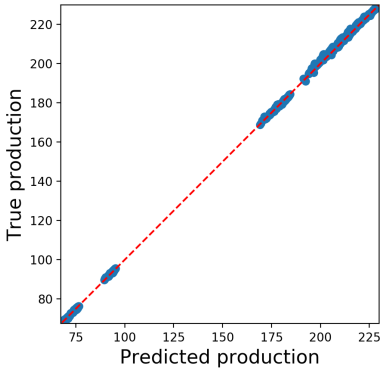


Figure 11.8: Comparison between the predicted optimal productions (x axis) and the true optimal productions (y axis) of the test set. The ideal predictions are shown with the dashed red line.



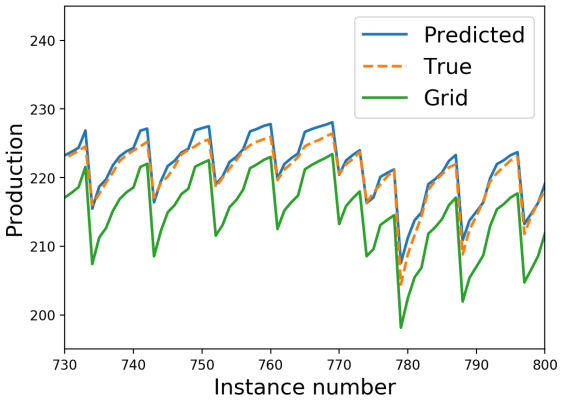
(a) Linear Regression

(b) Neural Network

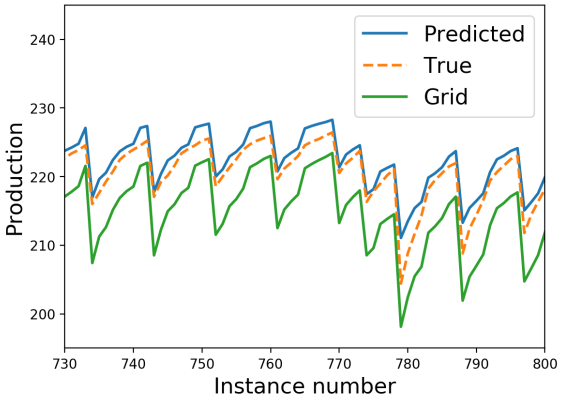


(c) Support Vector Regression

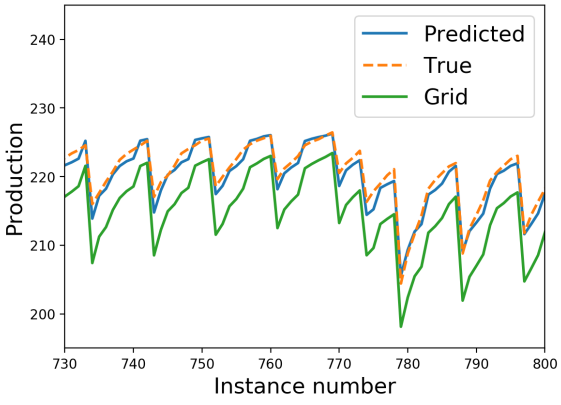
Figure 11.9: Power production (y axis) for different instances (x axis). In green the power production of a layout on a regular greed, in orange the power production (for the same site and turbines) obtained with an optimized layout. In blue the prediction from the different ML models (the closer to the orange line, the better).



(a) Linear Regression

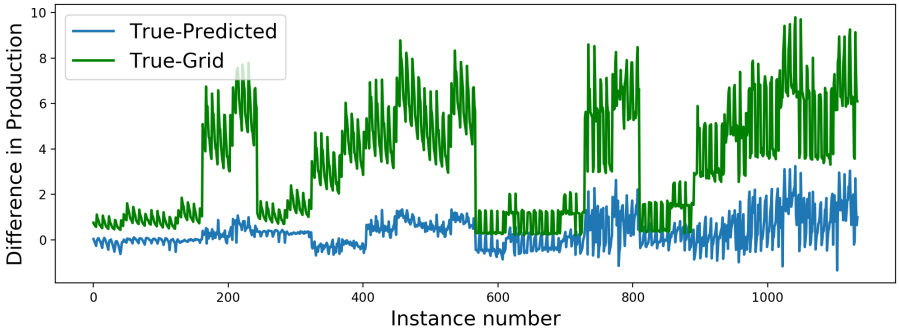


(b) Neural Network



(c) Support Vector Regression

Figure 11.10: Difference between true and grid production (in green) and true and predicted production for our SVR model (in blue). The comparison shows the importance of using ML instead of a simple grid evaluation, for a correct ranking of possible new sites.



Part VI

Closing

Chapter 12

Conclusions and future work

12.1 Conclusions

In this thesis we have shown how improvements in the design phase of a wind park can have a significant impact on the Levelized Energy Cost (LEC) of the project. Maximizing Annual Energy Production (AEP) and minimizing both immediate costs (CAPEX such as foundations and cables) and the life-time costs (as cable losses), allowed Vattenfall to have more competitive prices in the tender for new sites. In the bigger picture, decreasing the price of wind energy could make renewable energy more attractive; in the long run, this can help reducing CO2 emissions and so global warming.

In order to make wind energy competitive, every part of an offshore wind park must be optimized to improve efficiency and to reduce costs. In this thesis we have shown how Mathematical Optimization can significantly improve several steps of the design phase. In particular we have addressed turbine allocation, inter-array cable routing, and optimization of jacket foundations.

As to the wind farm layout problem, our goal was the design of a fast heuristic capable of handling instances with 10 000+ possible positions in a matter of minutes. To this end, we have exploited two basic tools: a fast ad-hoc

heuristic, and a MIP model designed for the very large instances of interest. A synergic use of these two tools has been proposed, following a MIP-and-refine recipe where two different variants of the underlying MIP model have been solved through a proximity search heuristic. Computational results on a testbed of medium-to-large scale instances have shown that the approach outperforms a standard use of the two basic tools. A lesson learned is that the choice of the MIP model to be used is a critical step in the design of the overall heuristic framework, because an effective compromise between tightness and compactness is required. In particular, models that are considered weak when solving small instances to proven optimality can become effective when used in a refining mode for large instances. In this respect, we designed an original MILP model and its stochastic version to handle different wind scenarios. Our original work (Fischetti and Monaci, 2016) was extended to consider other requirements arising in practice, such as considering different turbine types, or nearby wind parks. Our overall solution method was tested on real-world sites, proving its value against commonly used commercial software. The tool is now routinely used within Vattenfall to design new parks. In particular, having a fast optimization tool allows the company to perform different what-if analyses to quantify the impact of alternative design choice.

Secondly, we have introduced a new MILP model for optimal cable routing in offshore wind farms. A main novelty of our model is its capability of taking both installation costs and power losses into full account. We have also developed a new matheuristic framework to have a practical optimization tool to be used on difficult real cases. Thanks to the collaboration between Vattenfall and DTU, we have been able to describe the problem as it appears in real applications, and to validate our results on real-world instances that have been made publicly available for benchmarking. Using a sound matheuristic framework, for most of our instances we have been able to produce extremely good solutions in about 15 minutes of computing time on a standard PC. We have shown that savings in the order of millions euros can be potentially achieved in the lifetime of a wind park. These kinds of highly-optimized layouts cannot be produced by a manual operator, due to the complexity of the corresponding design problem. In addition, evaluating the impact of losses from a long term perspective and understanding how the layout should be changed in order to reduce them, represents an extremely valuable analysis for the company, that could not have been carried out otherwise. We dedicated a full paper to quantify the impact of considering cable losses in real offshore cable routings. This analysis was done both comparing with existing layouts (such as the Horns Rev 1), and comparing different optimized solutions obtained considering different assumptions. In general, we observed that it is convenient to use cables with

less resistance in order to reduce power losses, even if those cables are more expensive at construction time. Nevertheless, it is very difficult to define some "rules-of-thumb" for this problem, since usage of cables and savings highly vary from instance to instance. This proves that a proper optimization tool, as the one we presented, is necessary for an optimal design of each layout. Finally, we considered new versions of the classical offshore wind farm cable routing problem. We have been able to investigate the most recent trends on the market and to evaluate their impact on the cable routing. Turbines are becoming more customized, allowing them to survive being disconnected from the grid in case of failures, or even to substitute substations through the so-called Offshore Transformer Modules (OTMs) (Siemens, n.d.). Turbine customization opens up for new possibilities in the park layout, therefore it is crucial to have an optimization tool able to quickly evaluate the economical impact of new technologies on the wind park costs. In this respect we have introduced a flexible and reliable optimization tool, that scales well for bigger parks and more complex constraints. We have been able to handle new features in the model (i.e., closed-loop structure, non-linear branch penalties, and OTMs) and to quantify their effect on real-world instances. The outcome of our tests indicates that millions of euros are involved in these analyses, so decisions based on optimized solutions can lead to substantial savings for the company and, more generally, to cheaper transition toward sustainable energy.

We have also started working on the optimization of jacket foundations. We have shown that MILP models can be used to better perform the tube selection for a given foundation structure, in order to minimize the amount of steel (and hence the price) of the structure itself.

Given the two faces of this project (the academic and the practical), we decided to have a whole part of this thesis dedicated to the usage of our tools in practice. In this part we have shown how our tools have been internally validated against the standard tools, and have provided examples of what-if analyses performed at design phase. We have also discussed the example of Danish Krieger Flak, where our optimization tools have been used in the bid that was recently won by Vattenfall with a record low price.

To conclude, we have also investigated the use of Machine Learning together with Mathematical Optimization to predict the optimal production of offshore wind parks. We have shown that ML techniques, trained on a large number of optimized solutions, can predict the optimal value of new instances of the same problem rather well. In this work, we have focused on the wind park layout problem. The exact value of optimized solutions for this specific problem is very difficult to compute, given the large number of constraints and non-linearities

involved, and can only be obtained using a sound optimization tool. In our tests we have shown that the ML estimate highly outperforms the human one. Being able to estimate the optimal value for a given optimization problem can be of interest by itself (as in the wind farm application studied in this paper), but can also be very useful, e.g., for heuristic node pruning in a branch-and-bound solution scheme.

All in all, the work presented in the thesis shows how relevant Mathematical Optimization is for a modern energy company.

12.2 Future work

As there still is a high demand to continue the reduction of LEC for wind energy, the work that has been presented in this thesis can be further expanded in different directions.

Looking at the wind farm layout optimization, for example, one can consider turbine loads. In wind parks turbines downstream suffer not only wake effect (loss of velocity in wind) but also extra turbulence. In this thesis, we managed to consider wake effect in an effective MILP-and-refine strategy, by pre-computing the pairwise interference between turbines. Pre-computing the loads in a similar way, was however not possible, as the turbulence experienced by a turbine depends on the entire layout (i.e., by the position of all the other turbines in the site) and cannot be reduced to a pairwise effect. As of today, a minimum distance between turbines is imposed in the optimization, to compensate for not having explicit loads in the optimization (the further the turbines, the lower the turbulence). In practice, higher loads decrease the lifetime of a turbine. It would be interesting to model the relation between loads and extra maintenance costs and to include this dimension in our optimization. The optimizer should then place turbines to balance between possibly increased production and shorter turbine lifetime.

Another interesting future work could be integrating wind park layout optimization with cable routing and maintenance costs. Our wind farm layout optimization tool, indeed, focuses on minimizing wake effect, and therefore tends to spread turbines all over the site. Having a very sparse layout, while increasing production, can also potentially increase cable costs (as longer cables must be used) or maintenance costs (as turbines may be further to reach by the maintenance vessels). However, both the wind farm layout model and the cable routing one are large size models. Defining a unique model, optimizing layout

and cables (and potentially the route of a maintenance vessel) simultaneously, may be a very hard task. We performed some preliminary tests imposing a maximum distance between turbines in our layout model, and then optimizing the layout. We evaluated the results, comparing the total LEC, for more or less spread layouts. Our preliminary tests showed that, for the kind of sites that are on the market today, placing the turbines closer to each other does not pay off: the AEP of a site, indeed, is an income that lasts 20 years, while the main cable costs are paid only once. We therefore decided not to further investigate the integrated model. Nevertheless, if available sites will become bigger in the future (for the same MW installed), some limitations on the total area used in the layout may become more relevant.

Looking at new sites on the market, it may be relevant in the future to consider more complex areas than the ones we consider today. Given the always increasing competition, indeed, sites with worse seabed or wind conditions may become more attractive, as they attract less companies at tender phase. Having more complex seabed conditions may impact, for example, cable routing. In our tool we consider a fixed installation cost for cables, independently of their location in the site. In the future, it may be interesting to include in the optimization some areas that, if crossed, require extra installation costs for bad seabed conditions. As to wind, in our optimization we consider the wind to blow in a predictable way (we register wind statistics from the last 10 years, and we design our layout based on that). This assumption is nowadays true in most cases. Looking at the future, however due to climate change and global warming, more and more sites may decrease their predictability. It would be interesting to include some more uncertainty in our objective function, looking for a robust layout under small changes in the wind statistics.

Some future work could also be dedicated to use MILP techniques to further optimize turbine foundations. Our preliminary work on tube selection in jacket foundations showed a big potential in this application. An interesting (and more challenging) application could be to optimize not only the tube selection, but the whole structure (optimizing the number of joints and their connection).

Finally, future work could address optimization models for the onshore case. Onshore wind parks have some major differences compared to the offshore ones, for example the wind is not blowing uniformly in the site due to the presence of hills, forests, etc. Our interference computation should therefore be adapted accordingly. Also, the cable routing constraints can be very different. Inter-array cables onshore are, for example, located underground, so digging costs should be added. Due to these costs, parallel cables (laying in the same ditch) are preferred onshore. In addition, onshore cables should be located parallel

to existing roads. Finally, one should also consider the impact that the wind farm would have on the population living nearby. In particular, there are regulations imposing maximum sound levels originating from wind turbines at nearby houses, or other sensitive areas. The sound level depends not only on the distance between the wind farm and the sensitive areas, but also on the number of built turbines and their dimensions. As a consequence, also sound levels should be taken into account in the optimization.

Appendix A

Computing interference

Interference between turbines has to be taken into account when optimizing turbine position. The wake effect, indeed, can cause a strong reduction in the final power production. Note however that too complicated (and computationally heavy) models for the interference are impractical for large cases.
In this appendix the way we computed wake effect in our models is described.

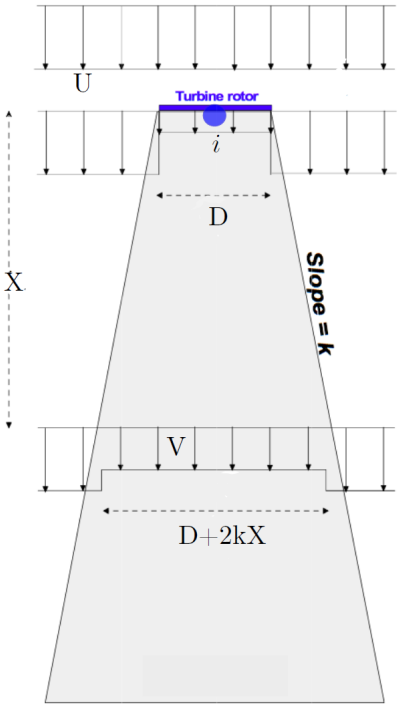
A.1 Jensen's model

The model proposed by N.O. Jensen Jensen, 1983 is a simple way to describe the wake effect between two turbines. This model is also used in WindPROEMD, n.d., an industry-standard software for wind resource assessment and placement of wind turbines within wind farms (and our reference software in the test phase of the model).

Let us consider two generic turbines, say i and j, such that i is the upwind one and j is the downstream one with respect to a specific wind direction. Let U denote the speed of the wind that affects i. Jensen's model describes the

wake effect behind i as a trapezium that has, as its minor base, the rotor of the turbine i of dimension, say, D (see Figure A.1)

Figure A.1: Wake effect according to Jensen's model with wind from north to south



Each point inside the trapezium (gray zone in the figure) is affected by the wake due to the presence of i. The loss of wind speed for a turbine (in the trapezium) at distance X from i, can be computed as

$$\delta V = U(1 - \sqrt{1 - C_t})(\frac{D}{D + 2kX})^2 \tag{A.1}$$

Where: U = upwind speed at turbine i
C_t = thrust coefficient corresponding to wind speed U

D = rotor dimension (diameter)
X = distance between i and j (on the x-axis)
k = wake decay constant, typically

$$k = \begin{cases} 0.075 & \text{onshore} \\ 0.050 & \text{offshore} \end{cases}$$

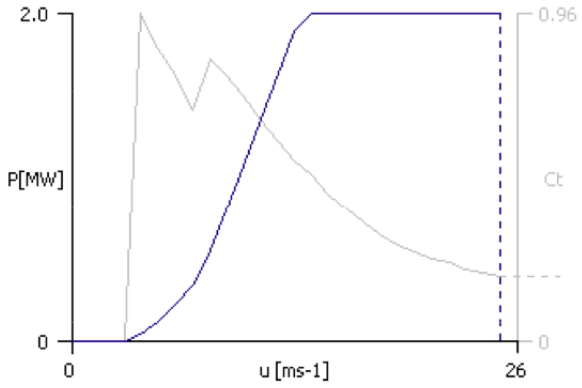
The wind speed at j is computed as $V = U - \delta V$. The interference $I_{i,j}$, defined as the loss of power on j due to i, can then be computed by using the turbine power curve (Figure A.3) according to the following table:

Figure A.2: Power and thrust coefficient C_t as a function of wind speed

Wind Speed (m/s)	Power (MW)	C_t
3.00	0.000	0.00
4.00	0.065	0.81
5.00	0.180	0.84
6.00	0.352	0.83
7.00	0.590	0.85
8.00	0.906	0.86
9.00	1.308	0.87
10.00	1.767	0.79
11.00	2.085	0.67
12.00	2.234	0.45
13.00	2.283	0.34
14.00	2.296	0.26
15.00	2.299	0.21
16.00	2.300	0.17
17.00	2.300	0.14
18.00	2.300	0.12
19.00	2.300	0.10
20.00	2.300	0.09
21.00	2.300	0.07
22.00	2.300	0.07
23.00	2.300	0.06
24.00	2.300	0.05
25.00	2.300	0.05

Note that both thrust coefficient C_t and power depend on the upwind speed in a non-linear way.

Figure A.3: Wind Speed vs C_t (gray line) and Wind Speed vs Power (blu line) Mortensen, 2010



The interference that a turbine causes using the Jensen's model has been plotted below. The upwind is supposed to come from north. Note that, when the upwind is 15.0 m/s, there is an effect of saturation in the power curve hence the difference of power (i.e. the interference) is really small (it goes from 0 to 0.01, see Figure A.4). Instead, when the upwind is 7.0 m/s the interference is stronger (it goes from 0 to 0.5, see Figure A.5)

Figure A.4: Interference and wind speed computed using Jensen's model; pixel = 10x10m; wind = 15.0 m/s from north

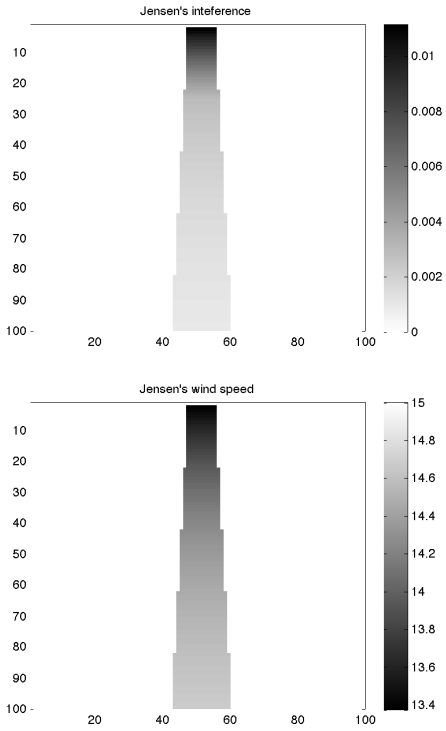
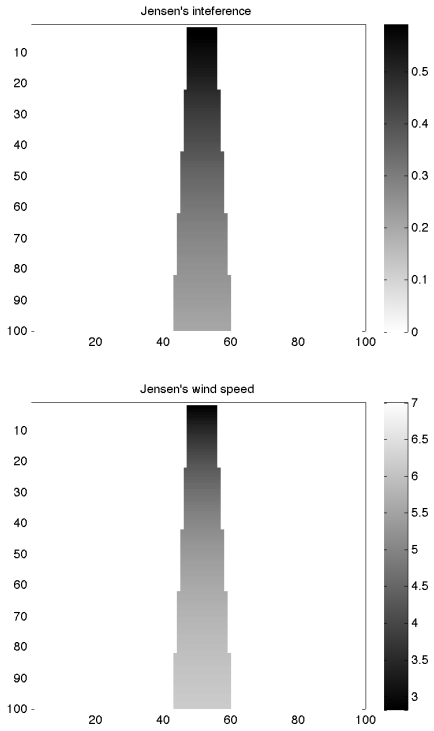


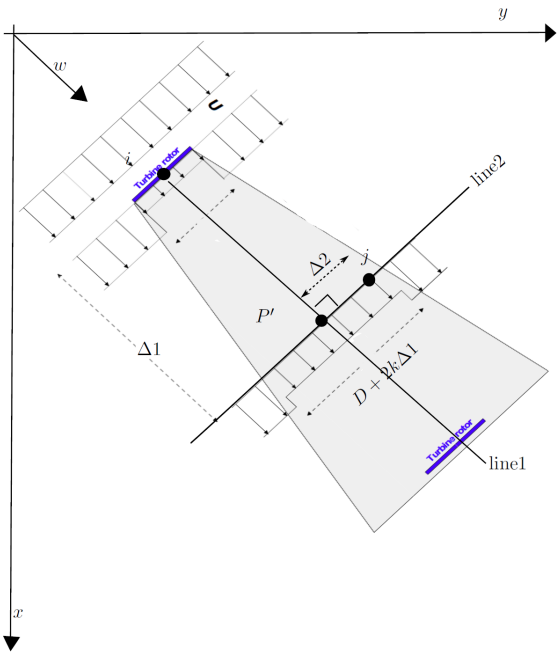
Figure A.5: Interference and wind speed computed using Jensen's model; pixel = 10x10m; wind = 7.0 m/s from north



A.2 Jensen's model for a generic wind direction

This section shows how to update Jensen's formula for a generic wind direction. In particular we need some easy calculations to know whether a given point j is in the wake produced by an upwind turbine i, when wind direction is not necessarily vertical. Consider then Figure A.6, where wind is not parallel to the x-axis. Note that the rotor automatically moves to become perpendicular to the wind direction.

Figure A.6: Wake effect modeled as in Jensen's model for wind coming from north-west



In the figure, w is a vector that describes wind direction, while $i, j \in \mathbb{R}^2$ are the two points where turbines are located. In order to decide whether j is in the wake of i, we need to determine the position of point P' in the figure, and to compute $\Delta 1 = \text{dist}(i, P')$ and $\Delta 2 = \text{dist}(j, P')$. Then j is in the shade trapezium iff $\Delta 2 \leq \frac{D+2k\Delta 1}{2}$.

In Figure A.6, the line that passes through i and goes in the wind direction is called line1 and is parametrized by α (say), while the line that passes through j and is perpendicular to line1 is called line2. P' is the intersection point between line1 and line2 so it satisfies both the equation that describes line1 (equation A.2) and the equation for the perpendicularity of the two lines (equation A.3):

$$rll \qquad P' = i + \alpha w \tag{A.2}$$

$$P' - i \perp j - P' \qquad \text{i.e. } (P' - i)^T (j - P') = 0 \tag{A.3}$$

Solving this system of equations leads to:

$$rll \quad \alpha = \frac{(j-i)^T w}{||w||^2} \tag{A.4}$$

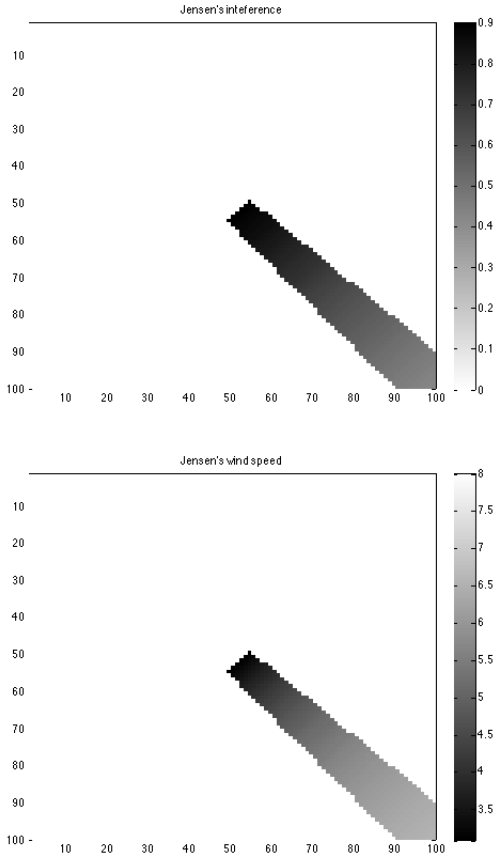
(A.5)

Given α, point P' can easily be computed by using equation (A.2).

Note now that if $\alpha \leq 0$ then i is upstream, so the interference on j due to i is zero (i.e. $I_{i,j}$=0). If $\alpha > 0$ instead we have to test if j is in the trapezium, i.e., if $\Delta 2 \leq \frac{D+2k\Delta 1}{2}$. In this case, interference can be computed according to formula (A.6), with $\Delta 1$ instead of X.

The plots in Figure A.7 as in Section A.1, with a wind direction $w = (0.5, 0.5)$ and speed 8 m/sec.

Figure A.7: Interference and wind speed computed using Jensen's model; pixel = 10x10m; wind = 8.0 from north-west



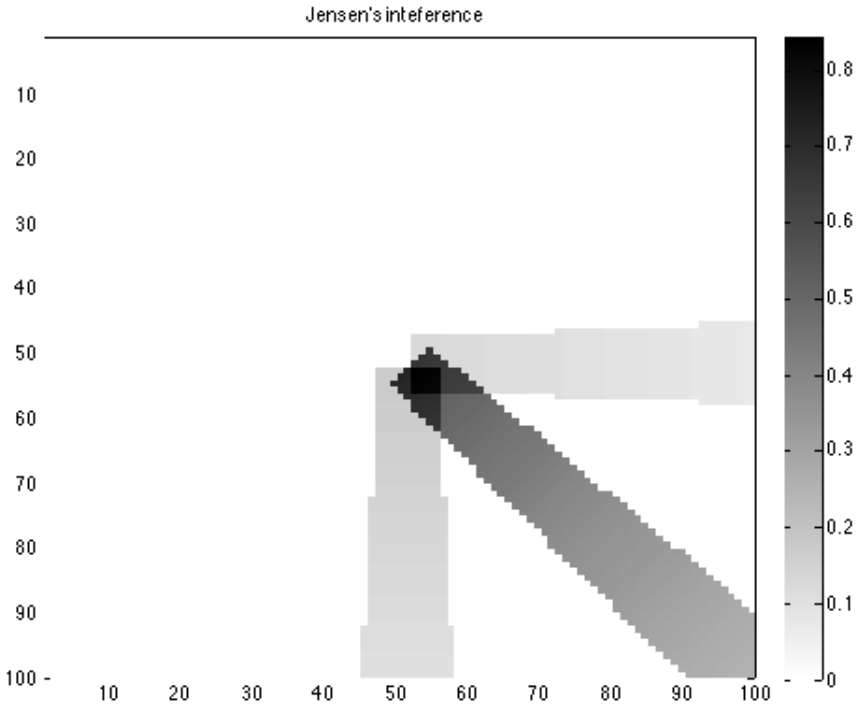
A.3 Weighted interference for the MILP models

Interference (computed as explained in the previous sections) will be used in the objective function of our mathematical models. Wind is however not stable and changes its direction and intensity over time. Therefore in our stochastic variant of the models we will use the average interference weighed by wind

probability.

Plots in Figure A.8 show the average interference in case of three possible wind situations:
1. wind (1.00, 0.00), speed 7 m/s, probability 0.30
2. wind (0.50, 0.50), speed 8 m/s, probability 0.60
3. wind (0.00, 1.00), speed 9 m/s, probability 0.10

Figure A.8: An example of $I_{i,j}$ computed as the average interference between three possible wind scenarios: wind = 7.0 m/s from north with probability 0.3, wind = 8.0 m/s from north-west with probability 0.6, wind = 9.0 m/s from west with probability 0.1 ; pixel = 10x10m;



If we would then plot the computed interference loss for real-world cases with all real-world wind scenarios, it would no longer look as a cone of interference,

but rather as a "star" of interference. The smaller the degree sampling, the denser the star. Figure A.9 shows the average interference calculated on real-world frequency series from the European Wind Energy Association EWEA (250,000+ real-world wind measurements). They have been divided in bins (24 sectors of 15 deg each for direction and in 1m/s bins for wind speed), obtaining about 500 macro-scenarios.

Figure A.9: Average interference with all different wind scenarios from the EWEA data

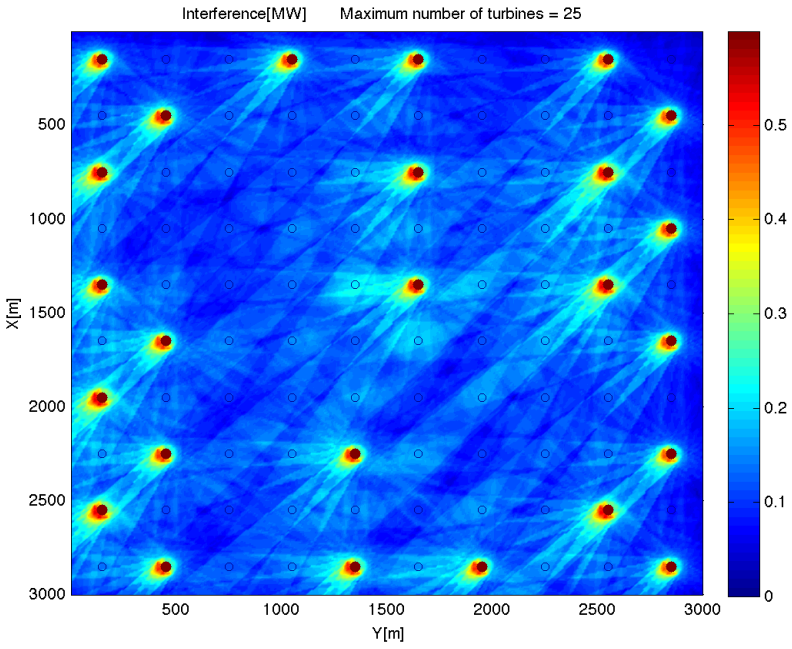


Figure A.9 represent a good visualization of the interference "star". Nevertheless, in our real-world studying in collaboration with Vattenfall, we further look into the definition of the sampling bins. It is indeed very important to properly define them, in order for our wake loss optimization to be meaningful.

Figure A.10: Considering only a few wind scenarios in input (every 30 degrees in the picture) results in a suboptimal layout.

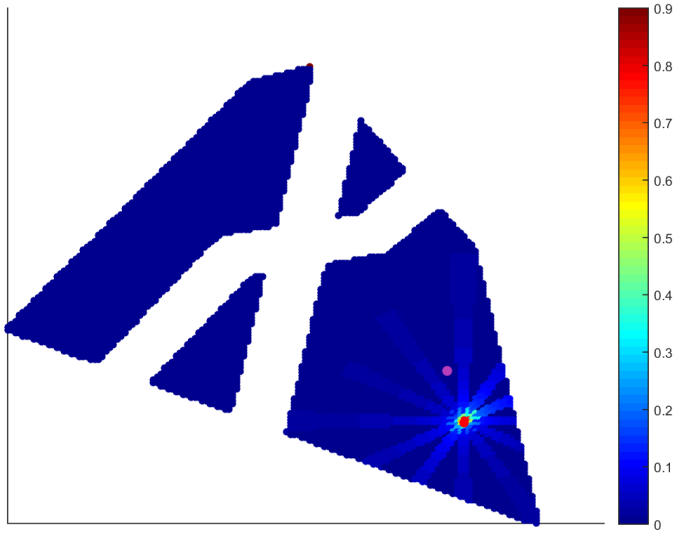
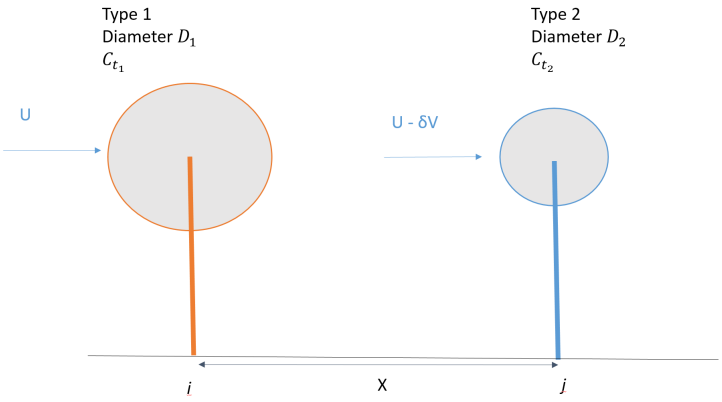


Figure A.10 gives a good example of how having only a few wind scenarios can be misleading for the optimizer. In this example, we binned all the real measurements in bin of 30 deg each for direction and in 1m/s bins for wind speed. According to our input, the mathematical model thinks that the wind blows only every 30 degrees. Therefore, the optimizer smartly puts turbines in between the cones (for example in the pink position of Figure A.10). This is, of course, wrong in reality. To avoid this kind of situations, we suggest to define bins of 0.1 degrees in practical applications. This of course means that the number of wind scenarios in the optimization increases drastically (we have 100 000+ wind scenarios in practical applications). Nevertheless, this proved not to be an issue, as we handle the interference computation in a pre-processing phase (according to our MILP model (4.13) – (4.18).

A.4 Wake effect between different turbine types

The wake effect computation should be adapted if multiple turbine types can interact. This is because different turbine models may have different rotor dimensions, different C_t and C_p curves, and different heights. In this case, we computed the loss in wind speed for each couple of turbines as follows.
Let us consider that the upwind turbine is of type 1 and the downwind one is of type 2. Each type has a specific rotor diameter (let us call it D_1 and D_2 respectively), a different thrust coefficient (let us call it C_{t_1} and C_{t_2} respectively) and a different power curve (let us call it P_1 and P_2 respectively). For the moment we assume the two turbines to have the same height. Figure A.11 schematically represents the case.

Figure A.11: Wake effect between two different types of turbine



The turbine in position j is affected by the wake due to the presence of a turbine in position i. The loss of wind speed for the downwind turbine at distance X from i, can be computed as

$$\delta V = U(1 - \sqrt{1 - C_{t_1}})(\frac{D_1}{D_1 + 2kX})^2 \tag{A.6}$$

Where: k = wake decay constant, typically

$$k = \begin{cases} 0.075 & \text{onshore} \\ 0.050 & \text{offshore} \end{cases}$$

Let us now consider the case in which turbines have different heights. The wind blows differently at different heights, so, depending on the turbine height, a turbine can experience more or less speed in the wind, in the same wind scenario. It is therefore important to include a scaling of the wind speed depending on the height, when we consider different turbines in the site.
We suppose that the wind in input U is the free wind registered at height h_0. The wind that is impacting turbines at different heights than h_0 needs to be scaled, we used the following scaling formula. For a generic turbine m of type k with height h_k, the scaling factor is:

$$hf_m = \frac{log(\frac{h_k}{z_0})}{log(\frac{h_0}{z_0})} \tag{A.7}$$

where z_0 is a roughness factor, that, for offshore cases, is fixed at 0.0002.
Let us consider now the interference between turbines at different heights. In this case, referring again to figure A.11, we say that turbine i (upstream), of type 1, has height h_1, while turbine j (downstream), of type 2, has a height h_2. When considering the case of interference between turbines the wind impacting the downstream turbine j is:

$$U_j = (U * hf_i - \delta V) * hf_{ij}$$

Where: U = is the free wind at default height (in input)
hf_i = is the scaling factor for the upwind turbine i computed as in (A.7)
δV = is the loss in wind speed as computed by the Jensen's model A.6
hf_{ij} = is the scaling factor considering both upstream and downstream turbines.

hf_{ij} is computed as

$$hf_{ij} = \frac{log(\frac{h_1}{z_0})}{log(\frac{h_2}{z_0})}$$

.

References

EMD (n.d.). *WindPRO*. http://www.emd.dk/windpro/frontpage.

Fischetti, M. and M. Monaci (2016). "Proximity search heuristics for wind farm optimal layout". In: *Journal of Heuristics* 22.4, pp. 459–474. ISSN: 1572-9397. DOI: 10.1007/s10732-015-9283-4. URL: https://doi.org/10.1007/s10732-015-9283-4.

Jensen, N. (1983). *A note on wind generator interaction*. Tech. rep. Technical Report Riso-M-2411(EN), Riso National Laboratory, Roskilde, Denmark.

Mortensen, N. G. (2010). *Wind farm AEP and wake loss calculations*. Tech. rep. Risø DTU, Denmark.

Siemens (n.d.). *New AC Grid Access Solution from Siemens: Lighter, faster, cheaper*. http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2015/energymanagement/pr2015030151emen.htm.